

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Diseño e implementación de una herramienta de autor para el
modelado y gestión de espacios**

Marcos Manzanares Diez

Tutor: Javier Gómez Escribano

Ponente: Germán Montoro Manrique

JUNIO 2016

Diseño e implementación de una herramienta de autor para el modelado y gestión de espacios

AUTOR: Marcos Manzanares Díez
TUTOR: Javier Gómez Escribano

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2016

Resumen

Cuando hablamos de personas con discapacidad cognitiva o discapacidad intelectual estamos hablando de aquellas personas con ciertas limitaciones durante su desarrollo y aprendizaje. Debido a esto, dichas personas terminan teniendo muchos problemas a la hora de enfrentarse en su vida a situaciones o actividades de la vida diaria de cualquier persona, como pueden ser, estudiar una carrera, conducir un vehículo o tener que ir al trabajo, es decir, de personas independientes. Al existir distintos grados de discapacidad, la persona puede ser más o menos independiente, pero con la ayuda de las nuevas tecnologías esta dependencia va siendo cada vez menor.

Una situación tan simple como orientarse en un espacio cerrado puede ser un impedimento para estas personas. Para solventar esta situación de una manera sencilla se ha creado una aplicación web que permite a las personas de apoyo crear mapas interactivos de espacios interiores con distintos modelos de mapa dependiendo del grado de discapacidad de cada alumno. Estos mapas van a servir de apoyo para que estas personas con cierta discapacidad puedan guiarse por los espacios interiores haciendo uso de la app móvil *AssisT-In*, que será el encargado de ofrecer el guiado.

Esta aplicación web se ha implementado usando tanto tecnologías como un diseño que nos permite adaptar la aplicación a distintos navegadores y a distintos dispositivos. Esto nos permite poder crear mapas *on-site* en una *tablet* o en un *smartphone*. La interfaz de usuario es una interfaz sencilla, clara y limpia, con el fin de facilitar la creación y modificación de los distintos mapas interactivos y sus elementos. Se ha usado como tecnología principal para la creación de mapas HTML5, debido al potencial que nos aporta poder trabajar con sus elementos *Canvas*. Para las consultas al servidor y su BBDD hemos usado el lenguaje PHP y como BBDD hemos usado SQL.

Palabras clave

Discapacidad cognitiva, espacios interiores, *browser*, HTML5, *Canvas*, dispositivos móviles.

Abstract

When we talk about people with cognitive disability or functional diversity we are talking about people with certain limitations during their development and learning. Because of this, those people turn out having a lot of problems coping in average day-to-day normal situations or activities, like go to university, drive a car or go to work, situations that could be normal if this people with cognitive disability were independents. Due to the existence of different grades of disability, a person can be more or less independent, but with the assistance of new technologies that dependence is lower than some time ago.

A simple situation like guiding themselves in an enclosed space could be a problem for people with functional diversity. To solve this situation in a simple way a web application has been developed. This application allows tutors of people with cognitive disability to create interactive maps of enclosed spaces with different models of the map depending on different grades of disability. This maps will help people with disabilities to guide themselves in to enclosed spaces using the smartphones app *AssisT-In*.

This web application has been implemented using technologies and a design that lets us to adjust the application to different browsers and different devices. This will allow us to create on-site maps in a tablet or a smartphone. The user interface is clean and simple, in order to enable the creation and modification of the different interactive maps and their elements. The principal technology that has been used is HTML5, due to the potential it gives us using their elements Canvas. For data base consults and server consults, it has been used PHP language for server consults and SQL for data base consults.

Keywords

Cognitive disability, indoor areas, browser, HTML5, Canvas, mobiles devices.

Agradecimientos

En primer lugar me gustaría agradecer todos estos años de apoyo y ayuda a mis padres, los pilares sobre los que he conseguido llegar aquí. Sin sus consejos, broncas, enfados y apoyos no hubiese conseguido llegar aquí.

Después a mis compañeros de la facultad, por aguantarme durante tantos años con mis cosas buenas y malas y por haberme apoyado y ayudado siempre que lo he necesitado.

También quisiera agradecerse a mis tíos y a mis primos que siempre que pueden me sacan una sonrisa y cuando se les necesita, siempre arriman el hombro.

Por último quisiera agradecerse a mi abuela, que falleció antes de que yo empezase este camino, pero que fue y es y seguirá siendo sumamente importante para conseguir cualquier cosa que me proponga.

INDICE DE CONTENIDOS

1	Introducción.....	5
1.1	Motivación.....	5
1.2	Objetivos.....	5
1.3	Organización de la memoria.....	6
2	Discapacidad intelectual.....	7
2.1	Definición.....	7
2.2	Causas de la discapacidad intelectual.....	7
2.3	Grados de la discapacidad intelectual.....	8
2.4	Proyectos para la integración de personas con discapacidad intelectual.....	8
2.5	Aportación del proyecto.....	9
3	Estado del arte.....	10
4	Análisis.....	13
4.1	Análisis de requisitos.....	13
4.1.1	Requisitos funcionales.....	13
4.1.1.1	Mapa.....	13
4.1.1.2	Elementos del mapa.....	14
4.1.2	Requisitos no funcionales.....	15
4.2	Metodología de desarrollo.....	15
4.3	Base de datos.....	16
4.4	Herramientas y tecnologías a usar.....	17
4.4.1	HTML, Javascript y JQuery.....	17
4.4.2	PHP.....	17
4.4.3	Bootstrap.....	17
4.4.4	XAMPP.....	18
5	Diseño.....	19
5.1	Arquitectura cliente-servidor.....	19
5.2	Diseño de base de datos.....	20
5.3	Diseño de la interfaz.....	22
5.4	Diseño de la iteración con el usuario.....	26
5.5	Diseño de los scripts y distintos ficheros.....	28
6	Implementación.....	29
6.1	Introducción.....	29
6.2	Módulos de la aplicación.....	29
6.2.1	Mapas.....	29
6.2.1.1	Creación de mapas.....	29
6.2.1.2	Borrado de mapas.....	31
6.2.1.3	Carga de un mapa.....	32
6.2.1.4	Edición de mapas.....	33
6.2.2	Nodos.....	33
6.2.2.1	Creación de nodos.....	33
6.2.2.2	Borrado de nodos.....	34
6.2.2.3	Edición de nodos.....	35
6.2.3	Uniones/Adaptaciones al usuario.....	36
6.2.3.1	Creación de uniones/adaptaciones al usuario.....	36
6.2.3.2	Borrado de uniones/adaptaciones al usuario.....	38
6.2.3.3	Edición de uniones/adaptaciones al usuario.....	39
7	Conclusiones y trabajo futuro.....	41
7.1	Conclusiones.....	41

7.2 Trabajo futuro	41
Referencias	- 1 -

INDICE DE FIGURAS

ILUSTRACIÓN 1. PROTOTIPO DE INDOOR SURVEY.....	11
ILUSTRACIÓN 2. INTERIOR DEL MADISON SQUARE GARDEN.....	11
ILUSTRACIÓN 3. MODELO EN CASCADA CON RETROALIMENTACIÓN.....	16
ILUSTRACIÓN 4. ARQUITECTURA CLIENTE-SERVIDOR	19
ILUSTRACIÓN 5. MODELO DE BASE DE DATOS	20
ILUSTRACIÓN 6. MAQUETA DE LA PÁGINA INICIAL.....	22
ILUSTRACIÓN 7. MAQUETA CON UN MAPA SELECCIONADO	23
ILUSTRACIÓN 8. MAQUETA DE AÑADIR UN NODO.....	24
ILUSTRACIÓN 9. MAQUETA DE AÑADIR UNA UNIÓN	24
ILUSTRACIÓN 10. MAQUETA DE EDITAR UN NODO	25
ILUSTRACIÓN 11. MAQUETA DE EDITAR UNA UNIÓN	25
ILUSTRACIÓN 12. MAQUETA BASE DE ERROR/AVISO	26
ILUSTRACIÓN 13. DIAGRAMA DE SECUENCIA PARA CREAR UN MAPA	27
ILUSTRACIÓN 14. DIAGRAMA DE SECUENCIA PARA CREAR UN NODO	27
ILUSTRACIÓN 15. CÓDIGO DE BARRAÍZQUIERDA.PHP.....	29
ILUSTRACIÓN 16. CÓDIGO DE INSERTAR UN MAPA EN BASE DE DATOS	30
ILUSTRACIÓN 17. SENTENCIA SQL PARA INSERTAR MAPA EN BASE DE DATOS	30
ILUSTRACIÓN 18. CÓDIGO DE SUBIRIMAGENMAPASERVIDOR.PHP	30
ILUSTRACIÓN 19. CÓDIGO DE LISTARMAPASBD.PHP	31
ILUSTRACIÓN 20. CÓDIGO DE BORRARMAPACOMPLETO.PHP	31
ILUSTRACIÓN 21. CÓDIGO DE OBTENERMAPA.PHP	32
ILUSTRACIÓN 22. CÓDIGO DE OBTENERNODOS.PHP	32
ILUSTRACIÓN 23. CÓDIGO DEL MÉTODO GETCANVASCOORDS	33
ILUSTRACIÓN 24. CÓDIGO CANVAS PARA DIBUJAR UN NODO	34

ILUSTRACIÓN 25. CÓDIGO DE SUBIRIMAGENNODOSERVIDOR.PHP	34
ILUSTRACIÓN 26. CÓDIGO DE BORRARUNIONBD.PHP	35
ILUSTRACIÓN 27. CÓDIGO DE BORRARNODOBD.PHP	35
ILUSTRACIÓN 28. LLAMADA MEDIANTE MÉTODO POST DE JQUERY A OBTENERFOTONODO.PHP ...	35
ILUSTRACIÓN 29. CÓDIGO CANVAS PARA DIBUJAR UNA UNIÓN	36
ILUSTRACIÓN 30. CÓDIGO DE INSERTARUNIONBD.PHP	37
ILUSTRACIÓN 31. CÓDIGO DE SUBIRIMAGENUNIONSERVIDOR.PHP	37
ILUSTRACIÓN 32. CÓDIGO PARA OBTENER EL PUNTO DE LA CURVA DE UNIÓN ADAPTADA	37
ILUSTRACIÓN 33. CÓDIGO CANVAS PARA DIBUJAR LA UNIÓN ADAPTADA	38
ILUSTRACIÓN 34. CÓDIGO DE MÉTODO CURSORELIMINARUNION(EV)	39
ILUSTRACIÓN 35. CÓDIGO DE BORRARUNIONBD.PHP	39

1 Introducción

1.1 Motivación

Como ya hemos comentado en el resumen anteriormente, la discapacidad intelectual conlleva una serie de limitaciones en las habilidades que la persona que las está adquiriendo. Dichas habilidades son habilidades básicas en la vida diaria de cualquier persona y estas limitaciones pueden ser una barrera u obstáculo a la hora de actuar ante distintas situaciones.

Para estas personas con discapacidad son de vital importancia tanto sus familias como sus educadores o personas de apoyo. Estos educadores pasan muchas horas con las personas con discapacidad y por ello les ayudan a superar sus obstáculos día a día. Uno de esos obstáculos es el de la orientación y estructuración del espacio, y más en detalle, la orientación en espacios interiores. Este problema de orientación viene dado por una alteración de las capacidades motrices de cada individuo, es decir, el grado de motricidad de cada individuo es completamente proporcional a su nivel cognitivo. [1]

Con el fin de poder eliminar esta barrera que tienen las personas con discapacidad cognitiva, se ha ideado esta herramienta web, que permite a los tutores crear mapas interactivos que posteriormente y mediante el uso de una app móvil, serán interpretados con un *Smartphone* que llevará consigo cada persona con discapacidad. Debido al marco en el que está inmerso el proyecto, que es hacer más fácil la realización de las tareas diarias, podríamos hablar de que esta herramienta web es tecnología de asistencia. Posteriormente explicaremos los distintos niveles de tecnologías de asistencia que hay y en cual está encuadrado nuestro proyecto.

En el laboratorio de Inteligencia Ambiental (*AmILab*) de la Universidad Autónoma de Madrid se desarrolla el proyecto *AssisT*, el cuál proporciona asistencia a personas con discapacidad cognitiva. Dentro de este proyecto hay otro proyecto denominado *AssisT-Out*, que es el que vamos a definir y explicar en este documento.

1.2 Objetivos

El objetivo principal de este proyecto es el de ayudar y permitir a los educadores poder crear mapas interactivos y adaptados a la capacidad de cada usuario, que permitan a estos ser lo más independientes posible. Para lograr esto nos hemos marcado una serie de objetivos a cumplir por nuestra aplicación web y que se describen a continuación:

- La facilidad de uso o **usabilidad** es uno de los objetivos primordiales en este proyecto. Los usuarios de esta aplicación pueden ir desde educadores que estén acostumbrados a navegar por Internet hasta educadores que no usen Internet frecuentemente, por lo que la aplicación web tiene que ser usable.

- Otro de los objetivos a lograr por la aplicación es la **robustez**. Es un objetivo principal de cualquier aplicación. Lograr este objetivo conlleva a tener una gran aplicación, ya que esta fallará poco y el usuario final podrá usarla sin problemas.
- Realizar pruebas con distintos educadores para poder contrastar opiniones y mejorar nuestra aplicación hasta el punto de conseguir una aplicación web cuyo mantenimiento es bastante sencillo y no requiere de proyectos evolutivos.
- Que los mapas creados y sus distintas adaptaciones a los usuarios se asemejen fielmente a la realidad del espacio interno que se quiere representar, ya que si no es así puede producir confusiones a las personas con discapacidad intelectual que usarán dichos mapas creados.

1.3 Organización de la memoria

La memoria del Trabajo Fin de Grado consta de los siguientes apartados:

- **Capítulo 1:** en este capítulo simplemente se hará una breve introducción a los contenidos de este proyecto y al trabajo realizado.
- **Capítulo 2:** se va a explicar de una manera resumida todo lo relativo a la discapacidad intelectual o cognitiva, cuáles son sus principales causas y sus distintos niveles.
- **Capítulo 3:** en este capítulo tratará de hacer un estudio del estado del arte, es decir, distintas tecnologías de asistencia que existen actualmente o han existido. Se intentarán realizar pruebas con ellas y poder sacar conclusiones que resulten interesantes para nuestro proyecto.
- **Capítulo 4:** en este apartado se va a realizar el análisis de nuestra aplicación web. Se explicarán los requisitos previos, tanto funcionales como no funcionales, las metodologías de desarrollo, el ciclo de vida de nuestra aplicación y las tecnologías que se han usado.
- **Capítulo 5:** se realiza una explicación más en detalle de toda la aplicación. Tanto arquitectura del sistema como diseño de la BBDD y diseño de la interfaz.
- **Capítulo 6:** en este apartado se explica todo lo relacionado a la implementación del proyecto. Se detallan tantos los distintos módulos de la aplicación como los lenguajes y las librerías usadas.
- **Capítulo 7:** se van a enumerar y explicar las distintas pruebas realizadas.
- **Capítulo 8:** en este capítulo se van a mostrar las conclusiones obtenidas con la realización de este proyecto. También se explicará el trabajo a realizar en un futuro sobre la aplicación.

2 Discapacidad intelectual

2.1 Definición

Según la American Association on Intellectual and Developmental Disabilities (A.A.I.D.D.) [2], la discapacidad intelectual o cognitiva podría definirse como:

“La discapacidad intelectual es una discapacidad caracterizada por limitaciones significativas tanto en el funcionamiento intelectual como en la conducta adaptativa, que abarca muchas habilidades sociales y prácticas cotidianas.”

Para que esta definición tenga sentido hay que tener en cuenta una serie de factores a la hora de valorar si una persona posee discapacidad intelectual o no:

- Las limitaciones de cada individuo deben compararse con el ambiente y la cultura de la que se encuentra rodeado cada individuo.
- Para que la evaluación de cada individuo sea consistente los educadores tienen que tener en cuenta la diversidad lingüística y las distintas maneras de comunicarse y comportarse debido a su cultura
- En estas evaluaciones que se realizan a los individuos se debe tener en cuenta las limitaciones de cada uno también coexisten con sus fortalezas, y debido a esto se pueden generar una serie de planes acondicionados a cada individuo que mejorarán la calidad de vida de dicho individuo si se mantiene el plan establecido durante un periodo de tiempo continuado.

Todo esto conllevará a realizar una buena evaluación de la discapacidad de cada individuo, determinar el grado de discapacidad de cada individuo, ir acabando poco a poco con sus limitaciones en el desarrollo de sus habilidades y que acciones cotidianas cada vez se realicen con una menor dificultad.

2.2 Causas de la discapacidad intelectual

La discapacidad intelectual afecta alrededor del 1 al 3% de la población mundial. Existen muchas causas que pueden provocar la discapacidad funcional, pero los médicos solo logran obtener la causa real en un 25% de los casos. Los factores de riesgo están directamente relacionados con las causas, que pueden dividirse en las siguientes categorías:

- **Infecciones:** estas pueden darse desde durante el embarazo hasta después del nacimiento del individuo. Algunas de ellas son la rubeola congénita, infección por VIH, encefalitis o meningitis.
- **Anomalías cromosómicas:** pueden ir desde un error en el número de cromosomas (síndrome de Down) hasta defectos en los cromosomas o en la herencia cromosómica (síndrome del cromosoma X-frágil o síndrome de Prader-Willi).

- **Ambientales:** en este caso la causa es el síndrome de privación.
- **Metabólicas:** como pueden ser el hipotiroidismo congénito, hipoglucemia o el síndrome de Reye.
- **Tóxicas:** en esta caso la discapacidad se puede deber a una exposición intrauterina al alcohol y las drogas o a una intoxicación por plomo o por metilmercurio.
- **Traumatismo:** la causa puede ser una hemorragia durante el embarazo o después del nacimiento, un traumatismo craneal grave o la falta de oxígeno en el cerebro del individuo.

2.3 Grados de la discapacidad intelectual

Los grados de discapacidad intelectual se definían hace no muchos años atendiendo al CI (Coeficiente Intelectual) de cada individuo. En la actual edición del DSM (Diagnostic and Statistical Manual of Mental Disorders) [3] se modifica esta manera de definir los grados de discapacidad y se acuerda la siguiente:

- **Leve:** el individuo puede necesitar cierto apoyo en las situaciones cotidianas de la vida diaria si se compara con sus coetáneos.
- **Moderado:** el individuo puede encargarse de sus tareas básicas como pueden ser comer, vestirse o su higiene personal pero para esto se requiere un periodo bastante extenso de enseñanza para conseguir cierta independencia en estas áreas. El individuo puede necesitar que se le recuerden ciertos aspectos sobre estas tareas.
- **Grave:** el individuo requiere supervisión durante todo el tiempo. Es incapaz de tomar decisiones sensatas sobre su bienestar y el de la gente que le rodea. En este tipo de grado el individuo requiere ayuda para realizar las tareas cotidianas que una persona con un grado moderado de discapacidad sí que es capaz de realizar.
- **Profundo:** en este grado el individuo depende de terceras personas para todos los aspectos básicos de la vida como pueden ser comer o mantener su higiene personal. Estas personas suelen presentar graves problemas tanto sensoriales como motrices.

2.4 Proyectos para la integración de personas con discapacidad intelectual

Las TIC (Tecnologías de la Información y la Comunicación) suponen uno de los avances más importantes en los últimos tiempos. Pero en ocasiones, estas TIC tan útiles en el día a día de las personas, puede que no estén acondicionadas y no sean accesibles.

Para conseguir la integración de estas personas en las TIC se llevan a cabo una serie de proyectos tanto nacionales como extranjeros. Alguno de los más importantes los explicamos a continuación [4]:

- **Proyecto *E-Inclusion*:** se trata del proyecto más importante y ambicioso llevado a cabo por las autoridades europeas. Este proyecto además de intentar que las TIC no excluyan a esas personas con discapacidad, también intenta que estas TIC mejoren su calidad de vida, a que mejoren la manera en la que se relación socialmente e incluso a encontrar puestos de trabajo.
- **Proyecto *TICTAC adapta*:** es un proyecto impulsado por el Ministerio de Industria que pretende la creación de una red de equipos informáticos en varias Comunidades Autónomas adaptados a las necesidades de las personas con discapacidad.
- **Portal *Discapnet*:** se trata de un portal creado por la Fundación ONCE y es una de las mejores herramientas que pone a disposición esta fundación para la integración de estas personas. Este portal proporciona a estas personas un acceso rápido y fácil a todo tipo de información que sea relevante tanto para ellos como para su entorno.

2.5 Aportación del proyecto

Como hemos comentado con anterioridad las personas con discapacidad cognitiva o intelectual pueden tener muchos problemas a la hora de orientarse debido al déficit producido por su discapacidad. Ese proyecto lo que persigue es facilitar la labor que realizan los educadores con estas personas cuando tienen problemas a la hora de orientarse en espacios internos.

Esta herramienta web pretende que los educadores puedan crear mapas interactivos basados en planos de espacios interiores que ayuden a estas personas. Estos mapas tendrán que ser bastante intuitivos ya que el público al que va dirigido esta herramienta puede ser muy diverso, con lo que tendrá que ser una herramienta intuitiva y sencilla. Esta herramienta mejorará tanto la vida cotidiana de las personas con discapacidad como la labor que ejercen sus educadores.

3 Estado del arte

En este apartado se va a realizar un estudio de tecnologías existentes en la actualidad, con las que poder obtener ideas y poder mejorar el resultado final de nuestra aplicación web. Para empezar, primero se debería explicar que es un *Sistema basado en localización* (LBS, Location Based Services).

Un **LBS** [11] son los servicios que buscan ofrecer a los usuarios un servicio usando su ubicación a tiempo real. Estos servicios pueden ser tanto para localizaciones al aire libre donde se pueden usar tanto las técnicas de localización para telefonía móvil como las técnicas de localización vía GPS como para localizaciones en interiores, donde las técnicas anteriormente mencionadas pierden calidad de servicio debido a los obstáculos que se encuentran las señales que se usan para la localización. En el caso que ocupa a este proyecto, vamos a explicar la localización en espacios interiores.

Estos sistemas de localización [12] en espacios interiores han sufrido una gran evolución desde su aparición hasta nuestros días. La primera tecnología que se creó con el fin de localizar a personas en espacios acotados hacía uso de los llamados marcadores fijos, es decir, marcadores reconocibles por una serie dispositivos distribuidos por todo el espacio interior seleccionado.

Un ejemplo de este tipo de localización es la distribución por todo el espacio acotado de distintos códigos QR que son reconocibles por la cámara de un Smartphone. La principal desventaja de este sistema es que el usuario solo va a estar localizado cuando decida hacer uso del Smartphone y leer uno de los códigos QR que hay distribuidos, es decir, no es un sistema de localización como tal ya que el usuario no se encuentra localizado en todo momento.

Posteriormente aparecieron los sistemas inalámbricos, que mediante el uso de ondas electromagnéticas, obtenían la localización del usuario. Este sistema se basaba en el concepto de emisores y receptores. Los emisores eran unos sensores completamente estáticos que emitían señales y de los que se conocían sus ubicaciones de antemano y el receptor, que en este caso sería el objeto que tiene que ser localizado, tendría un receptor de señal inalámbrica.

Si consideramos que el número de *smartphones* en el mundo es casi el mismo que el número de personas, llegamos a la conclusión de que los sistemas de localización en interiores basados en Wi-Fi, tecnología que todo *smartphone* tiene, son bastante prácticos ya que no requieren de tecnología adicional. La principal desventaja que presenta es que muchas veces la red Wi-Fi no es capaz de llegar a todos los puntos de espacio interior con una señal potente.

También han aparecido los sistemas de navegación inercial (INS). Con la aparición de estos sistemas se acaba con la dependencia que había sobre las infraestructuras a usar para la localización. Estos nuevos sistemas usan una serie de sensores como pueden ser giroscopios (sensores de rotación) o acelerómetros (sensores de movimiento con los que obtienen la orientación del movimiento que se está produciendo, la velocidad y la posición de un objeto concreto).

Actualmente, unas de las aplicaciones por excelencia en cuanto a la localización en espacios interiores es *Indoor Survey* [13] de *Apple*. Esta aplicación permite la creación de mapas para el posicionamiento en interiores y la creación de puntos en mapas que usaremos para indicarle a la aplicación nuestra posición dentro del espacio interior. Esta aplicación nace de la adquisición de la compañía *WiFiSlam*, de Silicon Valley, por parte de *Apple*. *WiFiSlam* consiguió, mediante las señales de radiofrecuencia emitidas por los puntos de acceso del Wi-Fi, detectar la localización de un Smartphone dentro de un edificio.

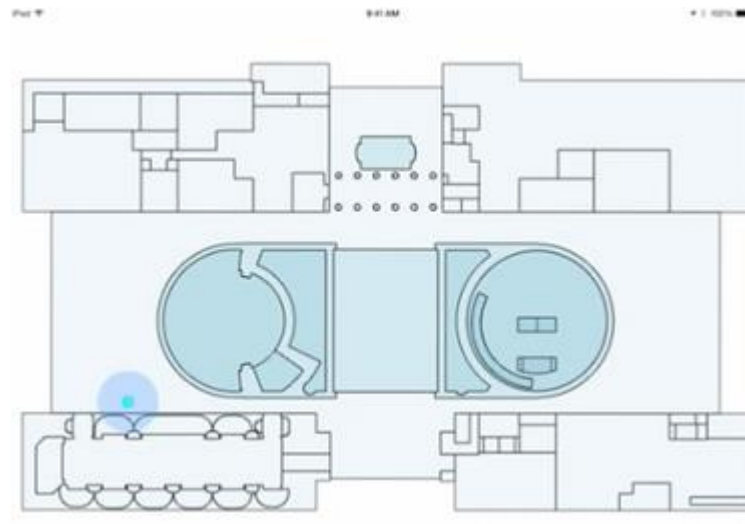


Ilustración 1. Prototipo de Indoor Survey

Google también se ha ganado su sitio en el mundo de la localización de espacios internos. En su aplicación *Google Maps* existe la posibilidad de visitar virtualmente el interior de una gran cantidad de edificios como pueden ser aeropuertos, centros comerciales o pabellones de congresos.

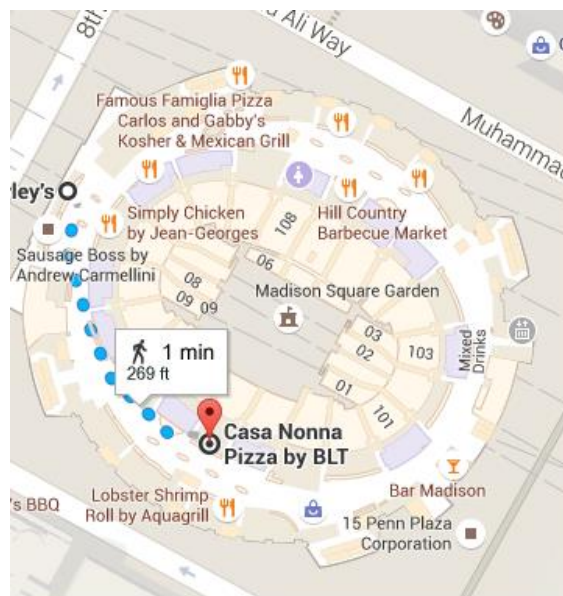


Ilustración 2. Interior del Madison Square Garden

4 Análisis

Esta fase de análisis y diseño es la fase inicial de todo proyecto software y una de las más importantes ya que el producto final depende en gran medida de esta fase. Durante esta fase se analizará el problema a resolver, de ese análisis se obtendrá el catálogo de requisitos que van a servir como base para las demás fases en las que se divide el proyecto, se va a definir la metodología a seguir durante el desarrollo e implementación del producto y se van a definir las herramientas y lenguajes a usar durante el proyecto.

4.1 Análisis de requisitos

En este apartado se va a realizar la toma de requisitos, tanto funcionales como no funcionales. En primer lugar se van a explicar ambos tipos de requisitos para después entrar más en detalle en los requisitos obtenidos y proceder a la explicación de cada uno de ellos:

- **Requisito funcional:** aquella característica del proyecto software que expresa una capacidad de acción o una función de dicho proyecto. Es decir, una funcionalidad de este.
- **Requisito no funcional:** este tipo de requisitos no tienen que ver con la funcionalidad del proyecto, si no con las características de funcionamiento del mismo.

4.1.1 Requisitos funcionales.

4.1.1.1 Mapa

- **RF01: crear mapa.** La aplicación web permitirá crear un mapa desde cero. Se mostrará una ventana emergente en la cual se podrá dar un nombre al mapa y seleccionar la imagen de dicho mapa. Estos mapas se mostrarán en una barra lateral situada en la parte izquierda de la pantalla.
- **RF02: borrar mapa.** También se podrá borrar cada uno de los mapas que tenga creado cada educador. La aplicación mostrará una *pop-up* de confirmación que preguntará si se desea borrar el mapa seleccionado. Si se decide borrar el mapa, se borrará tantos los datos de la BBDD como las imágenes del servidor relacionadas con este mapa.
- **RF03: modificar mapa.** La aplicación web también permitirá modificar un mapa ya creado. Para ello mostrará una ventana emergente idéntica a la que permite crear los mapas pero con los datos del mapa ya cargados. Lo único que se deberá hacer es modificar dichos datos.

- **RF04: generar código QR.** La herramienta web dará la opción de generar un archivo en formato *PDF* en el cual se mostrarán un código QR por cada nodo existente en el mapa. Estos códigos QR serán leídos por la aplicación para smartphones *AssisT-In*, que será la encargada de guiar a las personas con discapacidad intelectual por los espacios interiores.

4.1.1.2 Elementos del mapa

- **RF05: crear nodo.** La aplicación web permitirá crear nodos. Estos nodos son las distintas ubicaciones del mapa que visitará la persona con discapacidad intelectual. Para crearlos se mostrará un *pop-up* en el que se podrá dar un nombre al nodo, indicar si ese nodo va a ser nodo final o no y seleccionar la imagen para el nodo en caso de que dicho nodo se nodo final.
- **RF06: borrar nodo.** Se podrá borrar cada uno de los nodos existentes en el mapa que se tenga seleccionado. La aplicación mostrará una *pop-up* de confirmación que preguntará si se desea borrar el nodo seleccionado informando, además, de que si se borra el nodo se borrarán también todas las uniones y las uniones adaptadas relacionadas con ese nodo. Si se toma la decisión de borrar el nodo, se borrarán tantos los datos de la BBDD como las imágenes del servidor relacionadas con este nodo.
- **RF07: modificar nodo.** La herramienta web también permitirá modificar la información relativa a ese nodo. Para ello mostrará un *pop-up* idéntico al que se usa para crear los nodos pero en este caso cargará y mostrará toda la información de dicho nodo. Lo único que deberemos hacer es modificar dichos datos.
- **RF08: crear unión.** También se tendrá la opción de crear una unión entre dos nodos. En este caso se mostrará una ventana emergente que nos permitirá adjuntar dos imágenes, una por cada nodo.
- **RF09: borrar unión.** Se podrá también borrar las uniones del mapa seleccionado. La aplicación mostrará una *pop-up* de confirmación que preguntará si se desea borrar la unión seleccionada. Se borrarán tanto las imágenes del servidor relacionadas con este mapa como los datos de la BBDD.
- **RF10: modificar unión.** La aplicación web permitirá modificar las uniones creadas. Se mostrará una ventana emergente igual que la usada en la creación de las uniones pero con los datos ya cargados en esta ventana. Únicamente se deberá modificar dichos datos.
- **RF11: crear unión adaptada.** También se tendrá la opción de adaptar una unión entre nodos para un individuo en particular. Para esto primero se deberá seleccionar la persona para la que se desea adaptar la unión. Después se mostrará un *pop-up* en el que, al igual que en la creación de uniones, se deberán adjuntar dos imágenes.

- **RF12: borrar unión adaptada.** También se podrá borrar cada una de las uniones adaptadas. Al igual que ocurre en la creación de estas uniones, primero se debe seleccionar el usuario del que queremos borrar la unión. Esto borrará las imágenes del servidor relacionadas con esta unión adaptada y borrar la unión de la BBDD.
- **RF13: modificar unión adaptada.** La herramienta web permitirá modificar las uniones adaptadas a los usuarios. Se mostrará una ventana emergente idéntica a la que se usa para crear la unión adaptada pero con la información ya cargada de dicha unión. Lo único que se debe hacer es modificar los datos de la unión adaptada.

4.1.2 Requisitos no funcionales.

- **RNF01: rendimiento.** Lo principal en este sentido es que el tiempo de respuesta de las interacciones de nuestra aplicación web con el servidor, ya sea tanto cargando, borrando o modificando las imágenes, como realizando las consultas a la BBDD. Las interacciones con la BBDD son sencillas, con lo que el rendimiento debería ser acorde a lo que se requiere de nuestra aplicación.
- **RNF02: usabilidad.** La interfaz que se va a crear va a ser una interfaz simple a la par que atractiva. El modo de interactuar con la aplicación será muy sencillo. Esto supondrá que el educador que use la aplicación comprenda cuál es su funcionamiento de una manera fácil. En cuanto a la introducción de los distintos datos en la aplicación, esta deberá ser estructurada para evitar los posibles errores del educador.
- **RNF03: fiabilidad y seguridad.** Se conseguirá la integridad de todos los datos relativos a la aplicación ante cualquier imprevisto que pueda surgir como puede ser un problema en el ordenador en que se está ejecutando la aplicación o un fallo de conexión. Los datos de los educadores se cifrarán y se protegerán y no se facilitarán a ningún agente externo.
- **RNF04: mantenibilidad.** El modelo a seguir por la aplicación es un modelo vista controlador. Al seguir este modelo, todos los módulos de la aplicación están diferenciados unos de otros, con lo que facilita la solución de posibles errores y la implementación de evolutivos del proyecto.

4.2 Metodología de desarrollo

La metodología de desarrollo es un *framework* que se usa para planificar y estructurar el proceso por el cual se desarrolló en sistemas de información. Este *framework* consiste en herramientas, modelos y métodos para asistir al proceso de desarrollo de software. Estos *frameworks* suelen ser creados por distintas organizaciones que posteriormente promueven su uso.

Dentro de las distintas metodologías existentes hemos decidido optar por un modelo en cascada con retroalimentación, metodología de desarrollo bastante común y muy usada en el desarrollo software. Esto supone una mejora del modelo en cascada clásico, ya que si cuando se produce la toma de requisitos estos no logran definirse correctamente o alguno de estos requisitos es ambiguo siempre se puede volver a una fase anterior del proyecto y realizar las mejoras necesarias.

Este tipo de modelo de desarrollo consta de múltiples entregas al cliente con lo que conseguimos un *feedback* del estado de la aplicación por parte del cliente y cada entrega es mejor que la anterior. Después de varias entregas se conseguirá entregar la aplicación al cliente de acuerdo a las especificaciones acordadas con él.

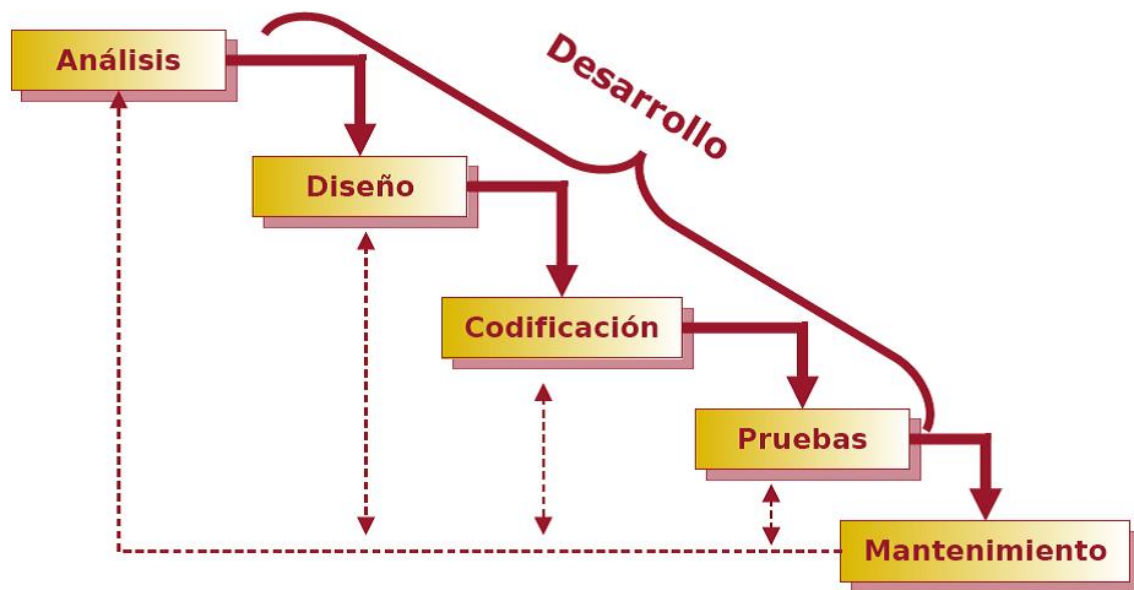


Ilustración 3. Modelo en cascada con retroalimentación

4.3 Base de datos

Como es obvio, la base de datos que se va a utilizar va a tener que estar situada en el servidor donde resida la aplicación web. En esta base de datos se van a almacenar los mapas con sus respectivos planos, los nodos con sus iconos en caso de que sean nodos finales, las uniones y adaptaciones al usuario y sus respectivas fotos, los tutores que hagan uso de la aplicación web y las personas con discapacidad cognitiva a las que vaya dirigido cada plano.

En el siguiente apartado de este documento se procederá a explicar más en detalle el diseño de la base de datos. En este apartado también se explicará el sistema de carpetas y subcarpetas que van a permitir poder tener todas las imágenes de los mapas perfectamente ubicadas dentro del servidor.

4.4 Herramientas y tecnologías a usar

A continuación se va a proceder a explicar las herramientas y tecnologías a usar para llevar a cabo este proyecto.

4.4.1 HTML, Javascript y JQuery

HTML, cuyas siglas significan *HyperText Markup Language* (Lenguaje de Marcado para Hipertextos), es el principal y más básico lenguaje de programación usado para la creación de una aplicación web. Este lenguaje se centra únicamente en el contenido de la página web, en ningún caso va a determinar su funcionalidad.

En el caso de este proyecto, vamos a usar la versión de HTML5 [6], debido a que cuenta con una característica denominada *Canvas* 2D. Esta característica se encargará de todo lo relativo a la creación, edición y presentación de los mapas en la página web, así como todos sus elementos (nodos, uniones entre nodos y adaptaciones de estas uniones).

Javascript [7] se trata de un lenguaje interpretado y orientado a objetos. También es conocido como lenguaje de script para aplicaciones web. Se trata de un lenguaje que no va a ejecutarse en el lado del servidor (donde se ejecutan HTML o PHP por ejemplo) si no que se ejecuta en el lado del cliente, es decir, en el navegador donde se esté corriendo la página web. El uso de este lenguaje va a permitir crear una página con una interfaz muy dinámica.

JQuery [8] es una biblioteca o API de Javascript, es decir, un compendio de funciones y métodos que facilitarán mucho la interacción con HTML.

4.4.2 PHP

PHP [9], cuyo acrónimo recursivo es *Hypertext Preprocessor*, es un lenguaje muy popular en el desarrollo web en el lado del servidor. Se trata de un lenguaje de código abierto y que nos da una facilidad como es que se pueda ejecutar tanto código HTML incrustado en el código PHP, como viceversa. Es un lenguaje simple pero que ofrece infinidad de características que pueden ser interesantes para el proyecto.

Este lenguaje, en un principio, se va a usar para llevar a cabo todas las conexiones necesarias con el servidor, ya sea tanto para realizar modificaciones en la base de datos, como para insertar, borrar o modificar los archivos relativos a este proyecto que se encuentren en el servidor.

4.4.3 Bootstrap

Bootstrap [10] es un *framework* que fue creado en un principio por Twitter. Se trata de un conjunto de librerías **CSS** que incluyen una gran cantidad de elementos que pueden ser usados para la maquetación de la parte Front-End de una aplicación web. Este *framework* sigue en continuo desarrollo en un repositorio de **GitHub**.

Con Bootstrap es posible crear interfaces gráficas sencillas y limpias y aporta una gran capacidad de personalización a la página web debido a que cuenta con muchos botones, tipografías, cuadros, menús, etc. Además tiene un soporte casi completo para **HTML5**, que como hemos comentado anteriormente vamos a usar en el desarrollo de la aplicación.

4.4.4 XAMPP

XAMPP [11] es un servidor independiente de Apache. Esta distribución ya cuenta con un intérprete de PHP, incluye un servidor de base de datos *MySQL* acompañado de su gestor *phpMyAdmin*. Se ha elegido este servidor debido a la facilidad de instalación en un ordenador portátil. Esta herramienta permitirá implementar los módulos de nuestra aplicación e ir realizando pequeñas pruebas unitarias que facilitarán la tarea de implementación.

5 Diseño

En esta fase de diseño se va a proceder a definir la arquitectura, componentes, interfaces y sistema de datos de la aplicación web de acuerdo a los requisitos anteriormente explicados. Para conseguir alcanzar un buen diseño lo más práctico es dividir en módulos el problema para llegar a un gran grado de cohesión (módulos específicos) y un grado mínimo de acoplamiento (módulos independientes).

5.1 Arquitectura cliente-servidor

Para el diseño de la aplicación lo primero que se va a tener en cuenta es que toda la información (base de datos, interfaz gráfica, etc.) va a estar alojada en el servidor y se va a acceder a ella mediante llamadas http usando un cliente o *browser*. A esta arquitectura se la conoce como **arquitectura cliente-servidor**.

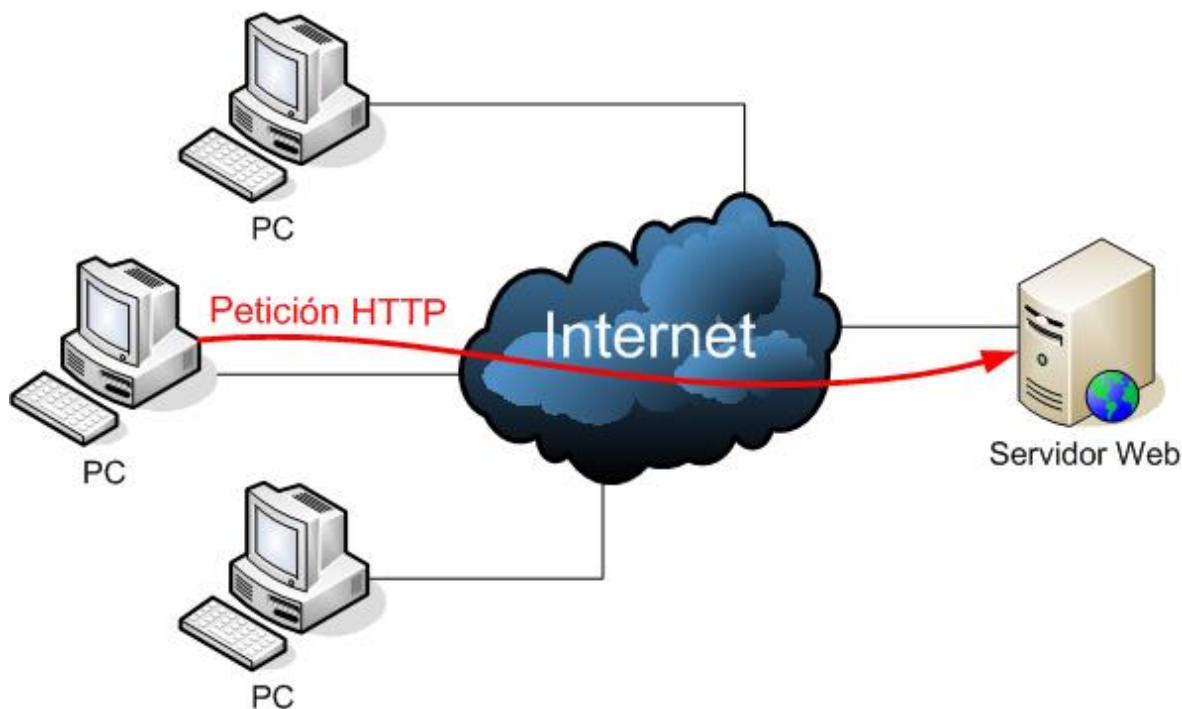


Ilustración 4. Arquitectura cliente-servidor

Como se puede observar en la imagen anterior es el cliente quien comienza con las comunicaciones enviando vía http peticiones al servidor, con lo que el cliente es una parte activa en las comunicaciones. En este tipo de proyectos, el cliente interactúa a través de la interfaz gráfica de la aplicación, con lo que esta debe ser usable y sencilla.

En el otro extremo de las comunicaciones se encuentra el servidor. Este va a tener un papel pasivo, ya que va a estar esperando peticiones de los clientes. Tras recibir estas peticiones, el servidor se encarga de procesarlas y devolver los datos solicitados al cliente.

Las principales ventajas con las que cuenta este sistema es la centralización de los recursos y los datos, lo que supone la integridad para estos. De esta manera se puede controlar mejor el acceso a los datos y se puede proceder a su modificación de una manera muy sencilla. Otras ventajas de esta arquitectura son su escalabilidad y su encapsulación, ya que es posible añadir o reducir su funcionalidad a aplicación fácilmente.

5.2 Diseño de base de datos

En este apartado vamos a explicar más en detalle todos lo relativo al diseño que se ha creído conveniente para la base de datos de la aplicación. Debido a que este proyecto es una parte de un proyecto de mayor envergadura, la base de datos va a tener dos tablas que va a compartir con otros proyectos, como son la tabla *tutors* y la tabla *users* que se explicarán a continuación junto con el resto de tablas de la aplicación.

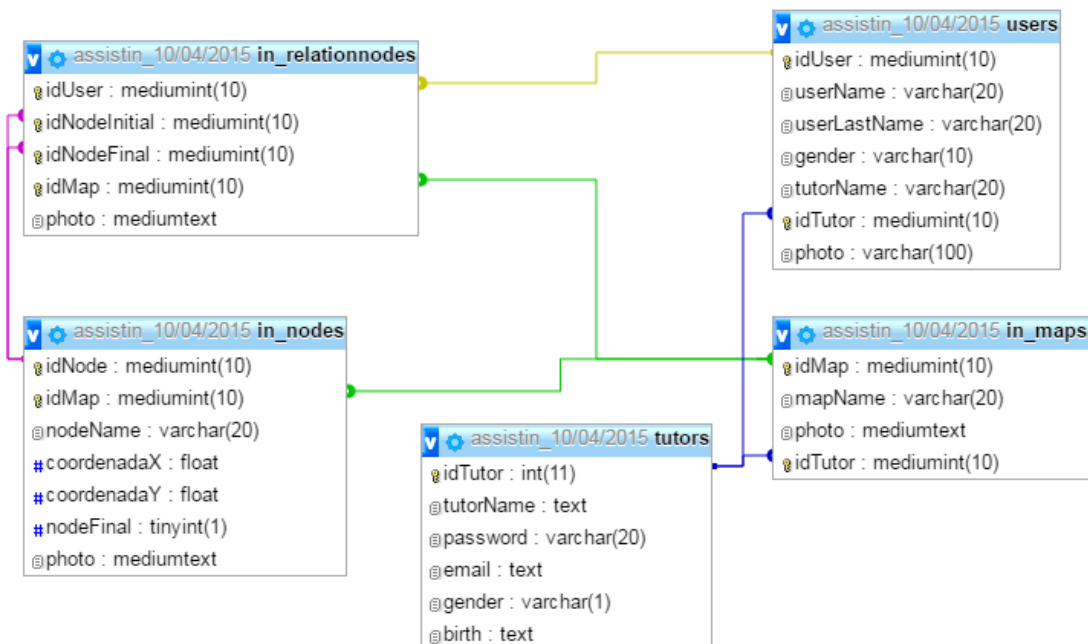


Ilustración 5. Modelo de base de datos

1. Tabla *in_maps*: en esta tabla se va a guardar la información relativa a todos los mapas de la aplicación. Esta tabla tiene como llave primaria tanto el identificar del mapa (*idMap*), como el identificador del tutor (*idTutor*), es decir, consta de una llave primaria compuesta. También va a guardar tanto el nombre que el tutor decida darle al mapa, como el nombre de la imagen que se va a guardar en el servidor relativa al mapa. Posteriormente detallaremos como se van a guardar en el servidor todas las imágenes.
2. Tabla *in_nodes*: esta tabla guarda todos los datos de cada uno de los nodos que se creen en la aplicación. Esta tabla también va a contener una clave primaria compuesta. Los dos datos que van a formar esta clave van a ser tanto el identificador del nodo (*idNode*) como el identificador del mapa (*idMap*).

También va a guardar el nombre del nodo (*nodeName*), el valor horizontal de la posición en el mapa (*coordenadaX*), el valor vertical de la posición en el mapa (*coordenadaY*), un valor numérico que os va a indicar si el nodo es de tipo nodo final, es decir, si va alguno de los caminos que cree el tutor va a finalizar en el nodo o no (*nodeFinal*) y que va a ser **1** o **0**, y el nombre de la foto del nodo (*photo*).

3. Tabla ***in_relationnodes***: esta tabla va a contener los datos de las uniones de los nodos y las adaptaciones de uniones entre nodos para los usuarios, es decir, va a haber dos tipos distintos de uniones; las uniones generales para todos los individuos con discapacidad cognitiva y las uniones adaptadas a cada individuo. El segundo tipo de unión va a estar adaptada a cada usuario dependiendo del grado de discapacidad funcional que tenga.

En ambos casos la uniones van a tener una clave primaria compuesta por cuatro datos: el identificador del mapa (*idMap*), el nodo inicial de la unión (*idNodeInitial*), el nodo final de la unión (*idNodeFinal*) y el identificador del usuario (*idUser*). En el caso de que la unión sea una unión general para todos los individuos el identificador de usuario que se usa es el valor **0** (usuario *Default*), mientras que en el caso de que la unión sea una adaptación, el identificador de usuario que se usa va a ser el identificador de dicho usuario en la tabla *users*. También se va a guardar el nombre de la foto del nodo de dicha unión (*photo*).

Cabe destacar que para cada unión vamos a tener dos entradas en la tabla. La primera entrada tiene como identificador de nodo inicial el valor de nodo final de la segunda entrada y la segunda entrada en base de datos tendrá como nodo inicial el valor de nodo final de la primera entrada. Por ejemplo, si vamos a guardar en base de datos la unión de los nodos **1** y **2**, en la primera entrada el valor de *idNodeInitial* es **1** y el de *idNodeFinal* es **2** y en la segunda entrada el valor de *idNodeInitial* es **2** y el de *idNodeFinal* es **1**. Se ha decidido optar por este método ya que así se puede tener en base de datos una distinción en cuanto a las fotos usadas para cada extremo de la unión.

4. Tabla ***users***: esta tabla va a contener la información de los usuarios que cada educador tenga a su cargo. La clave primaria va a ser compuesta y va a tener dos atributos; el identificador del individuo (*idUser*) y el identificador del educador (*idTutor*). También va a guardar el nombre (*userName*) y los apellidos (*userLastName*) de la persona con discapacidad funcional, su género (*gender*), una foto suya (*photo*) y el nombre de su educador (*tutorName*).
5. Tabla ***tutors***: esta tabla va a contener una clave primaria que va a ser el identificador del tutor (*idTutor*). También va a contener su nombre (*tutorName*), su género (*gender*), su fecha de nacimiento (*birth*), su email (*email*) y la contraseña con la que se va a logar (*password*).

5.3 Diseño de la interfaz

Como se ha mencionado con anterioridad, la aplicación debe ser una aplicación muy usable debido a que el rango de personas que puede usar esta aplicación es bastante amplio, es decir, va a ser usada desde personas que están familiarizadas con el uso de páginas web, hasta personas que la primera vez que consulten y usen una aplicación web sea con esta aplicación.

Se ha optado por diseñar una aplicación simple e intuitiva pero que a su vez tenga la capacidad de cubrir todos los requisitos especificados en los puntos anteriores. En este apartado se van a mostrar todas las maquetas que se han creado usando la aplicación de creación de maquetas *Balsamiq Mockups 3*.

La primera maqueta que se va a mostrar nos indica cómo se va a encontrar el usuario de la aplicación la pantalla según seleccione el apartado de crear mapas. Se ha optado por separar esta pantalla en dos partes muy diferenciadas. En la parte de la izquierda se van a mostrar todos los mapas que haya creado el tutor que se encuentre logueado. También se ha optado por que la creación y eliminación de mapas se realice desde esta zona de la izquierda.

En la parte de la derecha nos vamos a encontrar una barra que contendrá todo lo necesario para la edición de mapas y sus correspondientes elementos. También tendremos la posibilidad de crear los códigos QR en formato *PDF* de los nodos de un mapa.

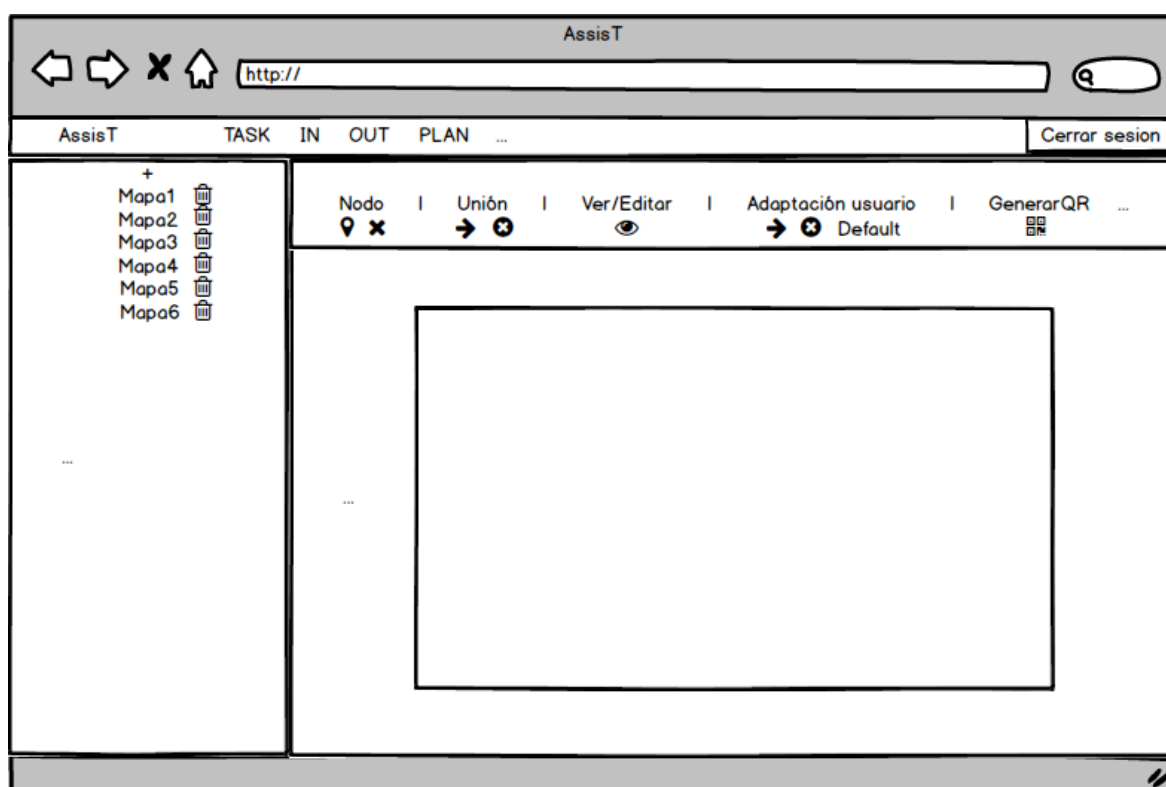


Ilustración 6. Maqueta de la página inicial

En la pantalla anterior, si se selecciona cualquiera de los mapas que ya tiene el usuario creado, se recargará la página y aparecerá el mapa cargado con sus uniones generales (líneas amarillas rectas) y sus nodos. En cuanto a las adaptaciones al usuario (líneas verdes discontinuas y curvas), para crearlas, eliminarlas o editarlas primero se deberá seleccionar el usuario pertinente en el apartado de *Adaptación usuario*, en la parte de *Default*.

Si pulsamos sobre *Default*, se nos abre un desplegable que tiene muestra a todos los usuarios del educador o tutor que esté haciendo uso de la aplicación. Es decir, en ningún caso vamos a mostrar las adaptaciones al usuario de todos los usuarios a la vez ya que se ha establecido que cada usuario puede tener adaptaciones distintas y lo único que conllevaría es a posibles errores del educador.

Posteriormente y para nos mostrar una cantidad repetitiva de pantallas de error/aviso, vamos a mostrar una pantalla que va a ser igual que las demás pantallas de error/aviso, pero la única modificación que hay entre ellas es el texto explicativo que contienen. En este caso, las pantallas de error/aviso se mostraran si se intenta seleccionar la característica de añadir o eliminar adaptaciones al usuario sin que haya un usuario seleccionado.

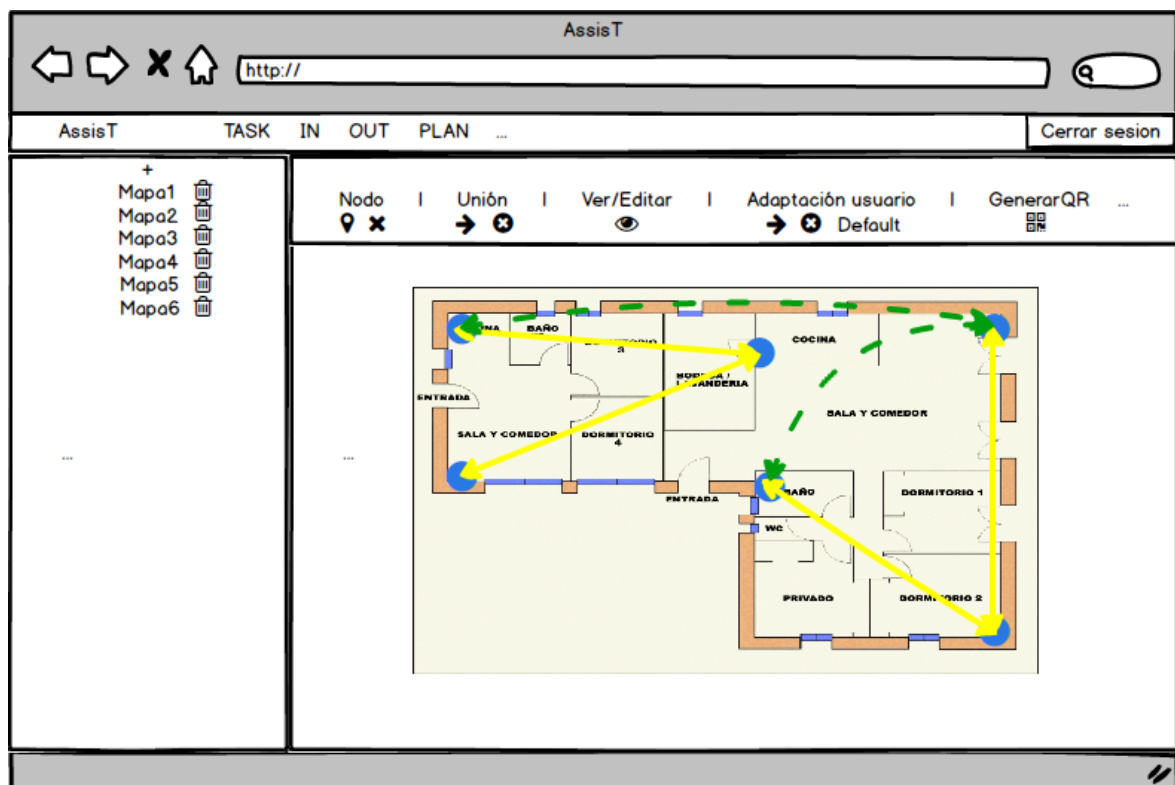
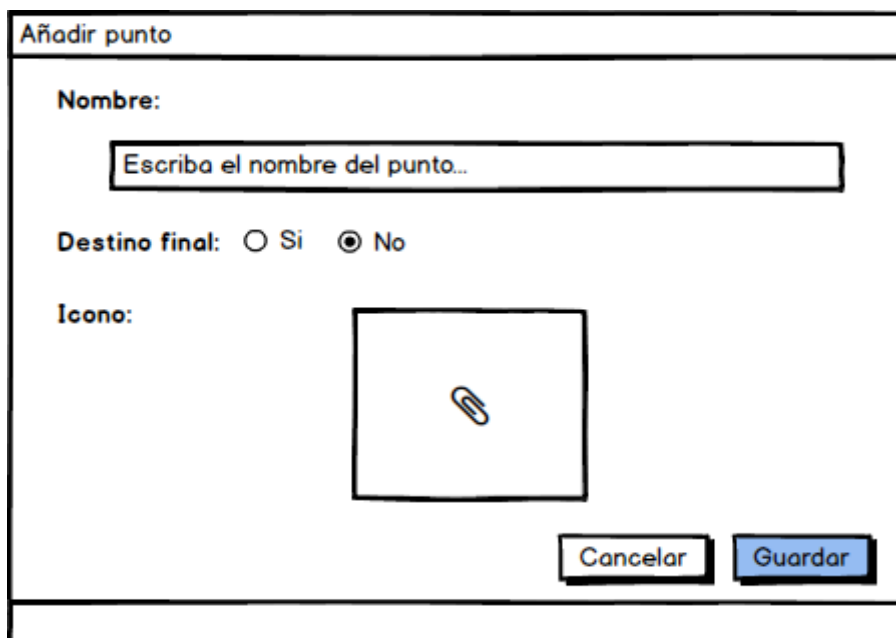


Ilustración 7. Maqueta con un mapa seleccionado

Si se opta por la opción de añadir un nodo al mapa, en la maqueta anterior en el apartado *Nodo*, va a aparecer la siguiente maqueta. En esta maqueta se deberá seleccionar un nombre para el nodo, tomar la decisión de si es *nodo final* o no y en caso de ser *nodo final*, añadir un icono a este nodo. En caso de que el nodo no sea *nodo final* la opción de añadir icono esta desactivada.



Añadir punto

Nombre:

Escriba el nombre del punto...

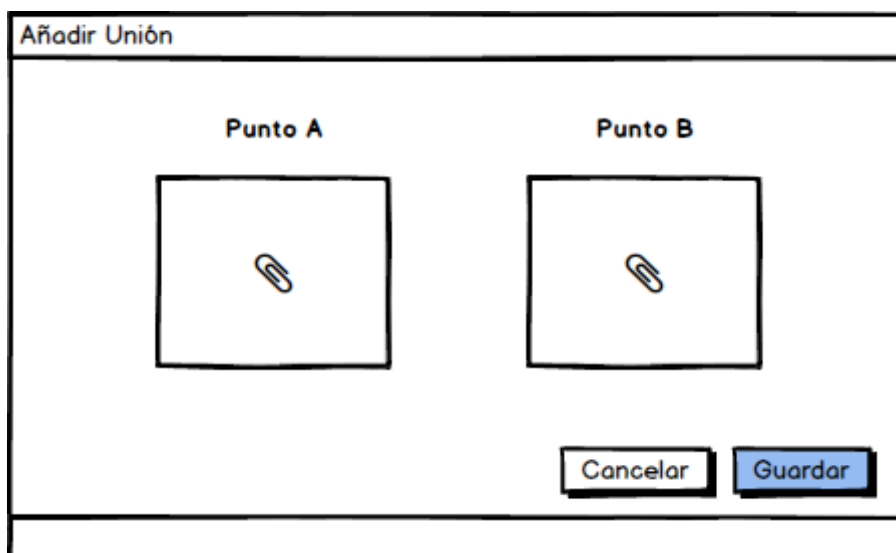
Destino final: ☐ Si ☒ No

Icono:

Cancel Guardar

Ilustración 8. Maqueta de añadir un nodo

Si por lo que se opta es por la opción de añadir una unión al mapa, en la segunda maqueta, en el apartado *Unión*, va a aparecer la siguiente maqueta. En esta maqueta que se muestra a continuación, vamos a tener que añadir un par de imágenes que nos muestren, desde un nodo hasta el otro nodo de la unión cual es el camino a seguir, es decir, una imagen panorámica en la que se pueda apreciar cual es la dirección a seguir para llegar hasta el siguiente nodo.



Añadir Unión

Punto A

Punto B

Cancel Guardar

Ilustración 9. Maqueta de añadir una unión

Si la decisión que se toma es la de editar cualquier nodo o unión, se va a tener que seleccionar la opción que está encuadrada en el apartado de *Editar/Ver*. Cuando tengamos seleccionada esta opción, va a ser posible editar o ver cualquier nodo, unión o adaptación al usuario únicamente pulsando sobre estos elementos. Es decir, si se pulsa sobre un elemento denominado *Impresora*, que fuese *nodo final* y que por lo tanto tenga un icono, la pantalla que aparecerá será la siguiente imagen que se muestra.



Añadir punto

Nombre:

Impresora

Destino final: ☒ Si ☐ No

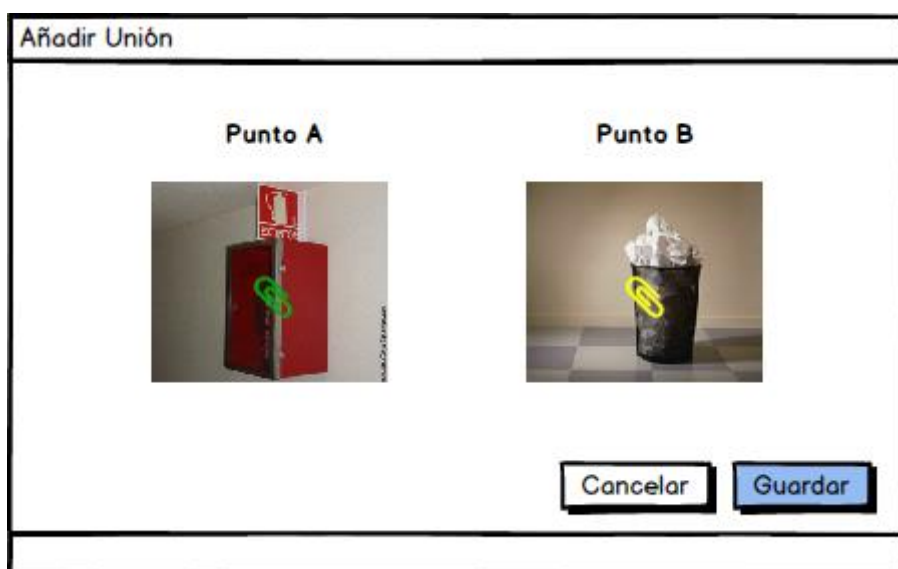
Icono:



Cancelar Guardar


Ilustración 10. Maqueta de editar un nodo

Si por el contrario se selecciona una unión o una adaptación al usuario, la pantalla que aparecerá será una pantalla que tenga las dos imágenes cargadas y que será como la siguiente maqueta que mostramos.




Añadir Unión

Punto A



Punto B



Cancelar Guardar

Ilustración 11. Maqueta de editar una unión

Por último, si se selecciona algún elemento cuando no se puede seleccionar, como puede ser añadir, eliminar o editar un nodo cuando no hay un mapa seleccionado todavía, o simplemente se desea guardar un elemento del mapa cuando su información no está completamente rellena, como puede ser el caso de que se intente guardar un nodo sin nombre o un nodo que sea considerado *nodo final* sin haber elegido un icono para este, aparecerá una pantalla como la que muestra la maqueta siguiente, en la que se describe el error obtenido.

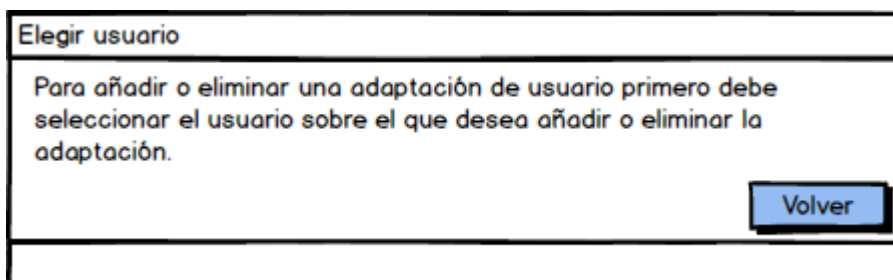


Ilustración 12. Maqueta base de error/aviso

5.4 Diseño de la iteración con el usuario

Antes de proceder a explicar la implementación que hemos seguido en cada módulo, se ha considerado importante dar una explicación mediante el uso de diagramas de secuencia de los pasos a seguir a la hora de crear un mapa y de crear un nodo. Estas dos acciones son bastante representativas del funcionamiento de la aplicación y mediante su explicación se pueden resolver dudas sobre el manejo de esta misma.

En el siguiente diagrama de secuencia muestra la manera de crear un mapa. Para seguir este paso, el educador que haga uso de la aplicación deberá estar debidamente logado en esta. Una vez hecho esto, tendrá que seleccionar el apartado de **In** en la barra superior de la aplicación. Ahora ya podrá crear el mapa.

Como se puede observar en la primera maqueta del apartado 5.3, en la zona izquierda donde se muestran los distintos mapas del educador hay en la parte superior un símbolo de +. Pulsando sobre el aparece una pantalla que permite añadir nombre y plano del mapa a crear. Una vez añadidos estos datos, si pulsamos sobre el botón **Guardar** los datos del mapa se guardarán en base de datos, el plano del mismo se guardará en el servidor y el mapa aparecerá en la zona izquierda.

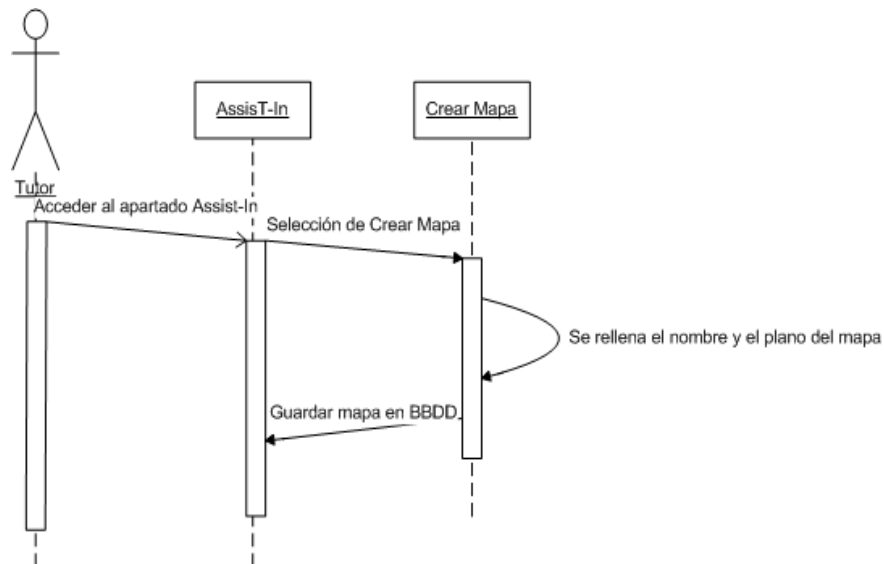


Ilustración 13. Diagrama de secuencia para crear un mapa

El segundo diagrama de secuencia que se va a explicar muestra los pasos a seguir para crear un nodo en un mapa ya seleccionado. En este caso, al igual que en el anterior caso explicado, el educador debe estar logado en la aplicación. Una vez logado, debe seleccionar el mapa sobre el que desea añadir el nodo.

Una vez que ya tiene el mapa cargado, debe pulsar sobre el icono de añadir nodo en el apartado **Añadir Nodo** de la barra de herramientas. Posteriormente, debe colocar el nodo en la posición del plano que desee y aparecerá una pantalla. En esta pantalla se deberá rellenar toda la información relativa al nodo, es decir, nombre de este, si se trata o no de un **nodo final** y en caso que sí lo sea, añadir un icono representativo del nodo. Una vez rellenada la información correspondiente, si se pulsa sobre el botón **Guardar**, se guarda en base de datos la información del nodo y en el servidor la imagen de este en caso de que se haya seleccionado una.

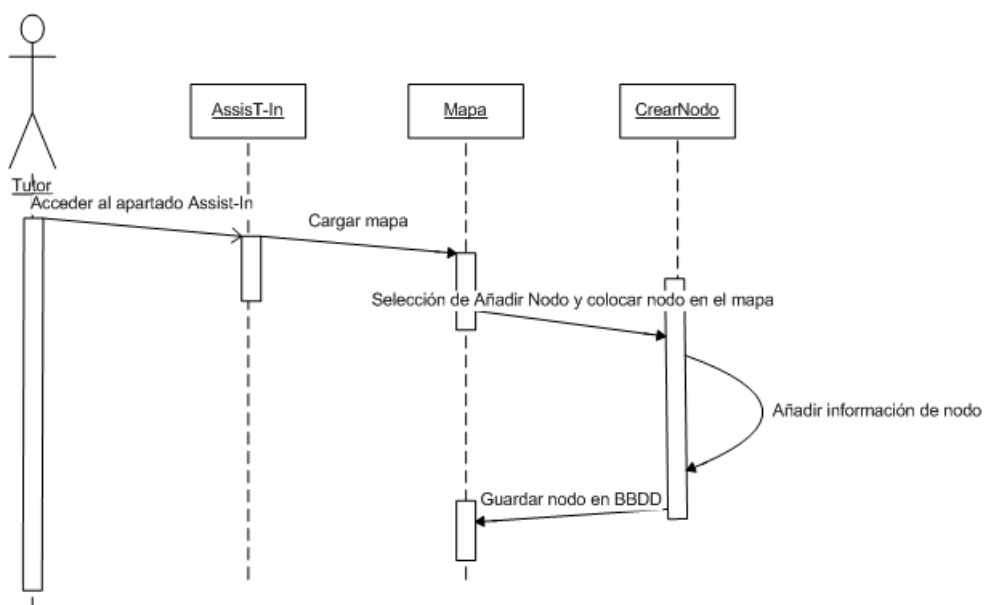


Ilustración 14. Diagrama de secuencia para crear un nodo

Como se puede ver en la imagen anterior, es muy sencillo crear un nodo ya que sigue una secuencia de pasos bastante simple. Para crear uniones o adaptaciones al usuario, simplemente habría que seguir los mismos pasos que para crear un nodo, pero eligiendo las opción de **Añadir Unión** en lugar de **Añadir Nodo** si queremos añadir una unión y eligiendo primero un usuario de la una lista sobre el que queremos adaptar la unión y después la opción de **Añadir Adaptación al Usuario** en lugar de **Añadir Nodo** si queremos añadir una adaptación al usuario.

5.5 Diseño de los scripts y distintos ficheros

Se va a dividir cada acción importante que requiera la aplicación en distintos archivos, cada uno de ellos bien diferenciado mediante su nombre, el cual indica a que acción va destinado dicho archivo. Siguiendo este procedimiento, el primer archivo importante que va a tener la aplicación es **index.php**. Este archivo va a contener la división de la pantalla en parte superior (**barraMenu.php**) y parte inferior. Esta parte inferior va a estar dividida a su vez en parte izquierda (**barraIzquierda.php**) y parte derecha (**barraHerramientas.php**).

En cuanto a la parte izquierda de la pantalla, **barraMenu.php** va a realizar una llamada a **listarMapasBD.php** que se va a encargar de rellenar la parte izquierda de la pantalla con los distintos mapas que el educador que se encuentra logueado tenga creados.

La parte derecha de la pantalla la va a gestionar **barraHerramientas.php**. Este archivo **php** va a contener el formato de todas las pantallas que hemos mostrado en el maquetado del punto anterior. También va a contener la carga del archivo **Funcionalidad.js**. Este archivo se va a encargar de toda la funcionalidad de la aplicación web. Va a encargarse de crear, editar y borrar los mapas y sus elementos (junto con los archivos **php** que a continuación se explican) y de crear el **pdf** con los **códigos QR** de cada nodo.

Como hemos comentado anteriormente, **Funcionalidad.js** se va a encargar de controlar la mayoría de las acciones que tiene la aplicación web. Esto lo va a hacer en compañía de una serie de archivos **php** que van a tener como principal fin, la conexión con la base de datos y las acciones relativas a esta. Algún ejemplo de estos archivos **php** son: **insertarMapaBD.php**, **insertarNodoBD.php**, **modificarImagenNodoServidor.php**, **subirImagenUnionServidor.php** o **listarUsuariosBD.php**.

6 Implementación

6.1 Introducción

En el desarrollo de este proyecto vamos a centrarnos en una las tecnologías más usadas por el ser humano, la web. En el año 2015, un 42% de la población mundial era usuario activo en internet. Esto nos permite que el número de usuarios que pueda tener nuestra aplicación web sea muy grande, que pueda ser usada desde cualquier dispositivo que tenga acceso a internet y en cualquier momento en el que sea necesario su uso. En este apartado se va a proceder a detallar los aspectos más relevantes de la implementación de la funcionalidad de cada módulo.

6.2 Módulos de la aplicación

A continuación se va a proceder a dividir la explicación de la implementación de la aplicación web en las distintas acciones que se pueden llevar a cabo para hacer más sencillo su entendimiento. Este apartado se va a dividir en: mapas, nodos, uniones, adaptaciones al usuario y códigos QR

6.2.1 Mapas

En cuanto a los mapas se van a poder realizar 3 distintas acciones: creación de un mapa, edición de un mapa y borrado de un mapa. La mayoría de estas acciones tienen llamadas al servidor para realizar acciones sobre la base de datos y sobre las imágenes que se almacenan en el servidor.

6.2.1.1 Creación de mapas

De la creación de los mapas se van a encargar los siguientes módulos: *barraIzquierda.php* para seleccionar la opción de crear un mapa, *barraHerramientas.php* que va a ser el módulo donde se aloje el código que muestra la pantalla donde introducir los datos del mapa, *Funcionalidad.js* que se encargará de recopilar los datos introducidos en la pantalla de creación del mapa y realizar un llamada al módulo *insertarMapaBD.php*, que va a guardar en base de datos la información del mapa y, solamente si esta llamada a base de datos se ha realizado correctamente, se llamará al módulo *subirImagenMapaServidor.php* que subirá la imagen del plano del mapa al servidor y con este finalizará la creación del mapa.

La siguiente imagen muestra el código de *barraIzquierda.php*, que como se puede observar va a realizar una llamada el método de *crearMapa()* del módulo *Funcionalidad.js*.

```
<script type="text/javascript" src="/js/Script1.js"></script>
<a style='cursor:pointer;' type="button" title="Cargar mapa"
  class="glyphicon glyphicon-plus" onclick="crearMapa()"
  style="float: center; width: 50%; margin:auto"></a><hr>
<?php include 'listarMapasBD.php'; ?>
```

Ilustración 15. Código de barraIzquierda.php

El método *crearMapa()* va a recoger toda la información necesaria para generar la entrada en base de datos y guardar la imagen en el servidor de la pantalla de crear mapa. Esta información necesaria va a ser el *nombre del mapa* y la imagen seleccionada como *plano del mapa*. También se va a recoger el identificador del educador que esté logado en la aplicación. Una vez obtenida esta información se va a realizar primero una llamada al módulo *insertarMapaBD.php* y posteriormente otra llamada al módulo *subirImagenMapaServidor.php* como se puede observar en la siguiente imagen.

```
$.post("insertarMapaBD.php", JSON.stringify(mapaJson), function() {
    var formData = new FormData();
    formData.append('archivo', fileMapa);

    $.ajax({
        url: 'subirImagenMapaServidor.php',
        data: formData,
        processData: false,
        contentType: false,
        type: 'POST',
        success: function(response) {
            var idMap = parseInt((JSON.parse(response)).idMap);
            mapaActual = idMap;
            mapaJson = {"nomMapa": nombreMapa, "idTutor": idTutor, "foto": foto, "idMap": idMap};
            arrayMapas.push(mapaJson);
            document.getElementById("fileOutputElem").style.backgroundImage =
                $('#fileOutputElemMapa').css("background-image");
            resetModalMapa();
            $("#sidebar-wrapper").load("barraIzquierda.php");
        }
    });
});
```

Ilustración 16. Código de insertar un mapa en base de datos

El módulo *insertarMapaBD.php* se va a encargar de recoger y tratar los datos que le llegan en formato JSON [11] y ejecutar una sentencia *INSERT* como la que muestra la siguiente imagen para guardar los datos del mapa en base de datos.

```
$resultado = $enlace->query("INSERT INTO in_maps (idMap, mapName, idTutor, photo)
VALUES ('$idMap', '$mapName', '$idTutor', '$photo');");
```

Ilustración 17. Sentencia SQL para insertar mapa en base de datos

En el módulo *subirImagenMapaServidor.php* va a tener lugar la subida al servidor de la imagen del plano del mapa en su carpeta correspondiente. Si esta carpeta no existiese, se procedería a su creación. La carpeta que se va a crear tiene un formato de nombre específico que van a seguir las carpetas relativas a cada mapa. El formato es *Mapa_id*, donde *id* es el identificador del mapa dentro de la base de datos.

```
if(!is_dir("Imagenes/")){
    mkdir("Imagenes/", 0777);
}
if(!is_dir("Imagenes/Mapa_$idMap/")){
    mkdir("Imagenes/Mapa_$idMap/", 0777);
}

move_uploaded_file($_FILES["archivo"]["tmp_name"], "Imagenes/Mapa_$idMap/.$fileName);
rename("Imagenes/Mapa_$idMap/.$fileName, "Imagenes/Mapa_$idMap/Mapa_$idMap$fileName);
```

Ilustración 18. Código de subirImagenMapaServidor.php

6.2.1.2 Borrado de mapas

En el borrado de mapas van a participar el módulo *listarMapasBD.php*, que mediante una consulta a base de datos va a mostrar en la zona izquierda de la pantalla un listado de todos los mapas creados por el educador que está usando la aplicación junto con un icono con forma de papelera, el módulo *Funcionalidad.js* que contiene el método *borrarMapa(idMap)* y que se encarga de mostrar la pantalla de aviso relativa a si se quiere borrar dicho mapa o no.

Si se selecciona que sí se desea borrar el mapa, se ejecuta la llamada al módulo *borrarMapaBD.php*. Este módulo va realizar el borrado en base de datos de todos los elementos que tengan que ver con el mapa que se ha decidido borrar y también elimina las carpetas del servidor relativas a dicho mapa.

En cuanto a *listarMapasBD.php*, este módulo únicamente va a realizar una consulta sobre base de datos para obtener los datos relativos a todos los mapas del educador y los va a mostrar todos en la zona izquierda de la pantalla.

```
echo "<p>".  
    "<a style='cursor:pointer;' onclick='cargarDatosMapa(" . $fila['idMap'] . ")'>"  
    . $fila['mapName'] . "</a>\n\n\n";  
    "<a style='cursor:pointer;' type='button' title='Cargar mapa' class='glyphicon glyphicon-trash'"  
    . "onclick='borrarMapa(" . $fila['idMap'] . ")'></a>".  
    "</p>";
```

Ilustración 19. Código de listarMapasBD.php

Como se puede observar en la imagen anterior, si pulsamos sobre el icono de *trash*, se va a realizar una llamada al método *borrarMapa(idMap)* del módulo *Funcionalidad.js*. Como se ha comentado anteriormente, este método va a mostrar un mensaje de aviso en el que se debe confirmar que se desea borrar el mapa. Una vez confirmado que se desea borrar se va a realizar una llamada AJAX al módulo *borrarMapaCompleto.php*.

El archivo *borrarMapaCompleto.php* va a borrar primero todas las entradas de las uniones y las adaptaciones al usuario relativas al mapa a borrar. Posteriormente borrará toda la información de los nodos pertenecientes a dicho mapa y por último borrará la entrada en base de datos del mapa en cuestión. Para finalizar con el borrado completo de la información en el servidor, se borrará la carpeta donde se almacena las imágenes relacionadas con el mapa mediante el uso de la sentencia *rmdir* de PHP.

```
$resultado = $enlace->query("DELETE FROM in_relationnodes WHERE idMap = '$idMap'");  
$resultado = $enlace->query("DELETE FROM in_nodes WHERE idMap = '$idMap'");  
$resultado = $enlace->query("DELETE FROM in_maps WHERE idMap = '$idMap'");  
rmdir("Imagenes/Mapa_$idMap");
```

Ilustración 20. Código de borrarMapaCompleto.php

6.2.1.3 Carga de un mapa

En la carga de los mapas el primer archivo que va a ser importante va a ser *listarMapasBD.php*, archivo del cual ya hemos hablado y que se encarga de cargar y listar los mapas del educador que esté usando la aplicación. Si en el caso de querer borrar un mapa, se debía pulsar sobre el icono de la papelera de dicho mapa, para cargar el mapa se debe pulsar sobre el nombre del mapa en cuestión.

Si se pulsa sobre un mapa se va a llamar a la función *cargarDatosMapa(idMap)* del archivo *Funcionalidad.js*. Esta función es la encargada de cargar toda la información relativa al mapa seleccionado, es decir, su nombre y su plano, toda la información de sus nodos, sus uniones y también va a cargar de la base de datos los datos de las personas con discapacidad cognitiva que trabajen con el educador.

En cuanto a la carga de datos de la base de datos se va a realizar mediante 4 llamadas AJAX. La primera de todas, *obtenerMapa.php* y que se encuentra en el método *cargarDatosMapa* va a devolver toda la información del mapa seleccionado incluyendo la imagen del plano seleccionado para este mapa.

```
$resultado = $enlace->query("SELECT * FROM in_maps WHERE idMap = '$idMap'");

$mapa = $resultado->fetch_array();
$photo = $mapa['photo'];
$fileType = substr($photo, strrpos($photo, ".") + 1);

header("Content-Type: image/$fileType");
$file = file_get_contents("Imágenes/Mapa_{$idMap}/$photo");

$miJSON = array('nombreMapa' => $mapa['mapName'], 'fotoName' => $photo,
                'fotoType' => $fileType, 'fotoURL' => base64_encode($file),
                'idTutor' => $mapa['idTutor']);
```

Ilustración 21. Código de obtenerMapa.php

La segunda, *obtenerNodos.php* está en *cargarNodos()* y va a cargar los datos de todos los nodos del mapa y la tercera y no menos importante es *obtenerUniones.php* que se encuentra en *cargarUniones()* y va a descargar del servidor la información necesaria de las uniones del mapa.

```
$resultado = $enlace->query("SELECT * FROM in_nodes WHERE idMap = '$idMap'");

$miJSON = [];
while ($node = $resultado->fetch_array()) {
    $photo = $node['photo'];
    $fileType = substr($photo, strrpos($photo, ".") + 1);

    header("Content-Type: image/$fileType");
    $file = file_get_contents("Imágenes/Mapa_{$idMap}/$photo");
    $miNodeJSON = array('nodeName' => $node['nodeName'], 'fotoType' => $fileType,
                       'fotoURL' => base64_encode($file), 'fotoName' => $photo,
                       'nodeFinal' => $node['nodeFinal'], 'x' => $node['coordenadaX'],
                       'y' => $node['coordenadaY'], 'idNode' => $node['idNode']);

    array_push($miJSON, $miNodeJSON);
}
```

Ilustración 22. Código de obtenerNodos.php

La última de todas las llamadas será la de cargar los datos de los usuarios con los que trabaja el educador, *obtenerUsuarios.php* y que se encuentra en el método *cargarUsuariosMapa()*. Con la carga de los usuarios finalizará el proceso de cargar un mapa.

6.2.1.4 Edición de mapas

Este apartado va a ser explicado tanto en el apartado en el que se va a explicar la creación, edición y borrado de nodos como en el apartado de creación, edición y borrado de uniones y adaptaciones al usuario.

6.2.2 Nodos

En esta sección se va a explicar todo lo relativo a la implementación de las acciones que se pueden llevar a cabo sobre un nodo. En primer lugar se explicará cómo es su creación, en segundo lugar como es su edición y para finalizar como se encuentra implementado el borrado de un nodo.

6.2.2.1 Creación de nodos

La creación de nodos es una de las principales acciones que se pueden realizar con la aplicación web. Para poder crear un mapa sobre el plano lo primero que se deberá hacer es seleccionar en la barra de herramientas (archivo *barraHerramientas.php*) la opción de creación de nodos. Una vez seleccionada esta opción, el siguiente paso es pulsar sobre el plano en el lugar en el que queramos crear un nodo. En este caso, se va a hacer uso de una función llamada *getCanvasCoords(x, y)*, la cual recibe como argumentos las coordenadas del *clic* que se ha hecho sobre el plano, es decir, las coordenadas basándose en alto y ancho de la pantalla y las transforma a coordenadas relativas al tamaño del plano.

```
function getCanvasCoords(x, y){  
    var canvasOffset = $("#fileOutputElem").offset();  
    return {"x": x - canvasOffset.left, "y": y - canvasOffset.top};  
}
```

Ilustración 23. Código del método *getCanvasCoords*

Posteriormente se nos va a abrir una pantalla que nos va a permitir añadir la información del nodo, es decir, nombre del nodo, si es *nodo final* o no y si es *nodo final*, se debe añadir un icono representativo del nodo. Una vez que se tengan los datos debidamente rellenados, si se quiere guardar el nodo se debe pulsar sobre el botón *Guardar*. Este botón se va a encargar de realizar una llamada a *dibujarPunto(punto, e)*. Esta función va a tener dos utilidades principales. La primera es la de dibujar en el plano el punto usando la tecnología *Canvas*.

```

var canvasCoords = getCanvasCoords(e.clientX, e.clientY);
var ctx = punto.getContext("2d");

ctx.fillStyle = "blue";
ctx.beginPath();
ctx.arc(canvasCoords.x, canvasCoords.y, 10, 0, Math.PI*2, true);
ctx.closePath();
ctx.fill();

```

Ilustración 24. Código Canvas para dibujar un nodo

La segunda utilidad va a ser la de guardar en base de datos la información que se ha introducido para el nodo y subir la imagen del icono del nodo. En caso de el nodo no sea *nodo final*, el icono que se subirá será un icono por defecto que usará en todos aquellos nodos que no sean nodos finales. Para guardar en base de datos se va a realizar una llamada a *insertarPuntoBD.php*, el cual únicamente lee los datos que se le han pasado mediante *JSON* y crea la entrada en la base de datos mediante una sentencia *INSERT*.

Para finalizar la creación del nodo, se va a proceder a subir la imagen de su icono al servidor. Para esto vamos a usar una llamada AJAX a *subirImagenNodoServidor.php* y la subida de la imagen se realizará como muestra la siguiente ilustración.

```

if($photo != "default.png"){
    $fileType = substr($_FILES["archivo"]["name"], strrpos($_FILES["archivo"]["name"], "."));
    $fileName = $_FILES["archivo"]["name"];
    move_uploaded_file($_FILES["archivo"]["tmp_name"], "Imagenes/Mapa_{$idMap}/".$fileName);
    rename("Imagenes/Mapa_{$idMap}/".$fileName, "Imagenes/Mapa_{$idMap}/Nodo_{$idNode}{$fileType}");
}else{
    $fileType = ".png";
    copy("Imagenes/default.png", "Imagenes/Mapa_{$idMap}/Nodo_{$idNode}{$fileType}");
    $resultado = $enlace->query("UPDATE in_nodos SET photo = 'Nodo_{$idNode}{$fileType}'
    | where idNode = '" . $idNode . "'");
}

```

Ilustración 25. Código de subirImagenNodoServidor.php

6.2.2.2 Borrado de nodos

Para poder borrar un nodo, se va a proceder primero a seleccionar en el barra de herramientas, en el apartado de *Nodo*, la opción de borrado de nodos. Esta opción está identificada con el icono de un aspa. Una vez seleccionada esta opción clicando sobre dicho icono ya se puede borrar cualquier que se encuentre en el mapa cargado. Para seleccionar el nodo a borrar solo se deberá clicar sobre el nodo en el plano mapa.

Este clic va a llamar al método *cursorEliminarPunto(ev)*, que lo primero que hará será transformar las coordenadas del clic que sean obtenido en coordenadas validas mediante el uso de la función *getCanvasCoords(x, y)* explicada anteriormente. Ya con las coordenadas transformadas, comprobará si dichas coordenadas coinciden con las coordenadas de algún punto del mapa. En caso afirmativo se mostrará una pantalla de aviso que advertirá al educador de que si borra dicho nodo, también borrará las uniones y adaptaciones al usuario relacionadas con ese nodo. Para borrar el nodo, solo se deberá pulsar sobre el botón *Si*.

Ahora la función *cursorEliminarPunto(ev)*, buscará y borrará tanto las uniones como las adaptaciones al usuario que determine que están relacionadas con el nodo que se ha decidido borrar. Para borrar las uniones de la base de datos, se realizará una llamada por cada unión a borrar a ***borrarUnionBD.php***, que va a eliminar las entradas en la base de datos correspondientes a dicha unión. Este procedimiento va a ser exactamente igual para eliminar las adaptaciones al usuario.

```
$resultado = $enlace->query("DELETE FROM in_relationnodes WHERE idNodeInitial = '$idNodeInicial'
AND idNodeFinal = '$idNodeFinal' AND idMap = '$idMap' AND idUser = $idUser;");

$resultado = $enlace->query("DELETE FROM in_relationnodes WHERE idNodeInitial = '$idNodeFinal'
AND idNodeFinal = '$idNodeInicial' AND idMap = '$idMap' AND idUser = $idUser;");

unlink("Imágenes/Mapa_{$idMap}/$photo1");
unlink("Imágenes/Mapa_{$idMap}/$photo2");
```

Ilustración 26. Código de borrarUnionBD.php

Para finalizar con el borrado del nodo, solo quedará borrar los datos relativos a este. Para esto, se va a realizar una llamada a ***borrarNodoBD.php***, que va a borrar la entrada de este nodo de la base de datos y la imagen que se había subido como icono del nodo.

```
$resultado = $enlace->query("DELETE FROM in_nodes WHERE idNode = '$idNode' AND idMap = '$idMap';");
unlink("Imágenes/Mapa_{$idMap}/$photo");
```

Ilustración 27. Código de borrarNodoBD.php

6.2.2.3 Edición de nodos

Para poder editar un nodo ya creado, lo primero que se deberá activar es la posibilidad de edición clicando sobre el icono de un ojo que se encuentra en la sección de *Ver/Editar* de la barra de herramientas. Una vez que tengamos esta opción seleccionada, solo se deberá clicar sobre el nodo que se desea editar al igual que cuando se quiere borrar un nodo.

Con este clic se va a llamar al método *editarContenidos()*. Este método es el que decidirá si se ha pulsado sobre un nodo y cargará toda la información del nodo en una nueva pantalla mediante una llamada al método *mostrarModalNodo(pos)*. Este método mostrará una pantalla exactamente igual que la que se usa para introducir la información cuando se quiere crear un nuevo nodo, pero con los campos rellenos por los datos correspondientes. Para poder cargar la imagen del nodo, se realizará una llamada a ***obtenerFotoNodo.php***, que descargará la foto solicitada.

```
$.post("obtenerFotoNodo.php", JSON.stringify(nodo), function(data) {

    var fotoURL = (JSON.parse(data)).fotoURL;
    var fotoTipo = (JSON.parse(data)).fotoTipo;
    URL = "url('data:image/" + fotoTipo + ";base64," + fotoURL + "')";
    $("#fileOutputElemPunto").css('background-image', URL);

});
```

Ilustración 28. Llamada mediante método post de JQuery a obtenerFotoNodo.php

Una vez que se tiene toda la información cargada en la pantalla de edición, se pueden editar los datos del nodo. Cuando se requiera el guardado de los datos, se deberá pulsar

sobre el botón *Guardar*, que llamará método *comprobarCambiosNodo(pos, URL)*. Este método se encargará de comprobar los cambios realizados y para finalizar con la edición del nodo, llamará a *modificarPuntoBD.php* y a *modificarImagenNodoServidor.php*. Estos módulos, a su vez modificarán la entrada pertinente en base de datos y la imagen correspondiente al nodo editado.

6.2.3 Uniones/Adaptaciones al usuario

Esta sección servirá como explicación de la implementación de las acciones que se pueden llevar a cabo sobre una unión o una adaptación al usuario. En primer lugar se explicará cómo se crean, en segundo lugar como se borran y para finalizar como se editan estas uniones. Se van a explicar ambos tipos a la vez aunque solo se hable de uniones ya que su creación es prácticamente igual, salvo en determinados aspectos que se comentarán más adelante.

6.2.3.1 Creación de uniones/adaptaciones al usuario

Para la creación de uniones deberemos seleccionar la opción de crear una unión en el apartado de *Unión* (en el caso de las adaptaciones al usuario el apartado será *Adaptaciones al usuario*) de la barra de herramientas. Una vez seleccionada esta opción ya se podrán seleccionar dos nodos para crear la unión entre ambos clicando sobre el nodo origen y sobre el nodo destino. Una vez llevados a cabo estos pasos, aparecerá una pantalla que nos va a permitir insertar una foto representativa del camino a seguir entre el nodo origen y el nodo destino y otra foto representativa del camino a seguir entre el nodo destino y el nodo origen.

Para guardar la información y una vez se hayan seleccionado las dos imágenes, se deberá pulsar sobre el botón *Guardar*. Al pulsar sobre este botón, se realiza una llamada al método *dibujarUnión(linea, ev)* que se encargará tanto de dibujar la unión sobre el plano usando la tecnología de HTML5 *Canvas*, como de guardar la información en la base de datos y las imágenes en el servidor.

```
ctx.lineWidth = 5;
ctx.strokeStyle = "yellow";
ctx.lineCap = "round";
ctx.moveTo(coordenadaX, coordenadaY);
ctx.lineTo(comprobarSiPunto.x, comprobarSiPunto.y);
ctx.stroke();
```

Ilustración 29. Código Canvas para dibujar una unión

Para guardar la información en la base de datos se realizará una llamada a *insertarUnionBD.php*, leerá los datos que se le pasan en formato JSON y creará dos entradas en base de datos, una que tendrá un nodo origen y un nodo destino y en la segunda entrada de base de datos el nodo origen será el nodo destino de la primera entrada y el nodo destino será el nodo origen.


```

$resultado = $enlace->query("INSERT INTO in_relationnodes (idUser, idNodeInitial,
                                                                idNodeFinal, idMap, photo) VALUES ('$idUser',
                                                                '$idNodeInicial', '$idNodeFinal', '$idMap', '$photo');");

$resultado = $enlace->query("INSERT INTO in_relationnodes (idUser, idNodeInitial,
                                                                idNodeFinal, idMap, photo) VALUES ('$idUser',
                                                                '$idNodeFinal', '$idNodeInicial', '$idMap', '$photo2');");

```

Ilustración 30. Código de insertarUnionBD.php

Este guardado en base de datos se realiza prácticamente igual tanto para las uniones como para las adaptaciones al usuario salvo que en el caso de las *uniones*, el valor de *idUser* será 0 y en el caso de *Adaptaciones al usuario* el valor de *idUser* vendrá determinado por el identificador del usuario para el que se ha hecho la adaptación. Para subir las imágenes seleccionadas al servidor se usa una llamada POST de *JQuery* a *subirImagenUnionServidor.php*. Este módulo se encargará también de renombrar convenientemente las dos imágenes a subir al servidor.

```

$fileType = substr($_FILES["archivo"]["name"], strpos($_FILES["archivo"]["name"], "."));
$fileName = $_FILES["archivo"]["name"];
move_uploaded_file($_FILES["archivo"]["tmp_name"], "Imagenes/Mapa_$idMap/.$fileName");
rename("Imagenes/Mapa_$idMap/.$fileName", "Imagenes/Mapa_$idMap/$photo1$fileType");

$file2Type = substr($_FILES["archivo2"]["name"], strpos($_FILES["archivo2"]["name"], "."));
$file2Name = $_FILES["archivo2"]["name"];
move_uploaded_file($_FILES["archivo2"]["tmp_name"], "Imagenes/Mapa_$idMap/.$file2Name");
rename("Imagenes/Mapa_$idMap/.$file2Name", "Imagenes/Mapa_$idMap/$photo2$file2Type");

```

Ilustración 31. Código de subirImagenUnionServidor.php

Una vez realizados estos pasos correctamente se tendrá creada la adaptación al usuario o unión de dos nodos. Existen dos diferencias que hacen que la creación de ambos elementos no sea exactamente igual. La primera diferencia reside en la manera de dibujar la línea, ya que la línea que se dibuja en la unión es una línea recta que consta de dos puntos marcados por el nodo origen y el nodo destino y la línea de la adaptación al usuario es una línea curva que consta también de un tercer punto que nos sirve como coordenadas de referencia para dibujar la curva.

```

var puntoIntermedioY = ((punto2.y + punto1.y)/2);
var puntoIntermedioX = ((punto2.x + punto1.x)/2);

if(((punto1.x - punto2.x) < 25) && ((punto1.x - punto2.x) > -25)){
    return {"x": (puntoIntermedioX + 50), "y": puntoIntermedioY};
}
else if(((punto1.y - punto2.y) < 25) && ((punto1.y - punto2.y) > -25)){
    return {"x": puntoIntermedioX, "y": (puntoIntermedioY + 50)};
}
else if((punto1.x < punto2.x) && (punto1.y < punto2.y)){
    return {"x": (puntoIntermedioX + 50), "y": (puntoIntermedioY - 50)};
}
else if((punto1.x > punto2.x) && (punto1.y > punto2.y)){
    return {"x": (puntoIntermedioX + 50), "y": (puntoIntermedioY - 50)};
}
else{
    return {"x": (puntoIntermedioX + 50), "y": (puntoIntermedioY + 50)};
}

```

Ilustración 32. Código para obtener el punto de la curva de unión adaptada

Este tercer punto se va a obtener mediante la llamada al método *crearPuntoCurva(punto1, punto2)*, el cual va a recibir como parámetros dos puntos con sus respectivas coordenadas. A partir de estos dos puntos se obtiene el punto medio del segmento que forman *punto1* y *punto2*. Una vez que tenemos las coordenadas de ese punto medio, lo único que se deberá hacer es modificar dichas coordenadas para obtener nuevas coordenadas que permitan crear las curvas de las adaptaciones al usuario. La manera de crear una línea curva usando tecnología *Canvas* también es diferente.

```
var puntoCurva = crearPuntoCurva({"x": coordenadaX, "y": coordenadaY},
                                  {"x": comprobarSiPunto.x, "y": comprobarSiPunto.y});
ctx.lineWidth = 5;
ctx.strokeStyle = "green";
ctx.lineCap = "round";
ctx.setLineDash([5, 20]);
ctx.moveTo(coordenadaX, coordenadaY);
ctx.quadraticCurveTo(puntoCurva.x, puntoCurva.y, comprobarSiPunto.x, comprobarSiPunto.y);
```

Ilustración 33. Código Canvas para dibujar la unión adaptada

La segunda diferencia apreciable es que para añadir una adaptación al usuario, primero se deberá seleccionar un usuario de la lista desplegable que se encuentra en la sección de *Adaptación al usuario*, ya que las adaptaciones al usuario tienen que ir unidas al usuario para el que se desea dibujar la unión adaptada. La aplicación no permitirá dibujar una adaptación si no se ha seleccionado un usuario de la lista antes.

6.2.3.2 Borrado de uniones/adaptaciones al usuario

La manera de borrar una unión o adaptación al usuario es sumamente parecida a la manera de borrar un nodo. El primer paso es seleccionar la opción de borrar una unión en el apartado de *Unión* (en el caso de las adaptaciones al usuario el apartado será *Adaptaciones al usuario*) de la barra de herramientas. Para seleccionar a unión a borrar, se tendrá que clicar sobre dicha unión.

Este clic va a llamar al método *cursorEliminarUnion(ev)*, que va a recorrer el array de uniones buscando la unión a borrar. Por cada unión que encuentre, va a obtener las coordenadas de los dos nodos que forman los extremos de la unión mediante dos llamadas a la función *devolverCoordsNodo(arrayUnionesN[i].nodo1)*, que recibe como parámetro uno de los dos nodos de la unión. Una vez ya tenemos las coordenadas de ambos nodos, se va a crear un rectángulo de tamaño semejante al del dibujo de la unión en el plano.

Este rectángulo va a ser la superficie sobre la que debe clicar el educador para seleccionar la línea a borrar y lo va a crear la función *crearRectanguloUnion(Coords1, Coords2)*. Esta función tiene dos parámetros que son las coordenadas obtenidas por las llamadas a la función anteriormente explicada. Para finalizar, se realizará una llamada a la función *dentroPoligono(arrayPolig, getCanvasCoords(ev.clientX, ev.clientY))*, que realizará la comprobación de si el clic del educador se encuentra dentro del rectángulo que nos indica la superficie de la unión.


```

var Coords1 = devolverCoordsNodo(arrayUnionesN[i].nodo1);
var Coords2 = devolverCoordsNodo(arrayUnionesN[i].nodo2);
crearRectanguloLinea(Coords1, Coords2);
var dentro = dentroPoligono(arrayPolig, getCanvasCoords(ev.clientX, ev.clientY));

```

Ilustración 34. Código de método `cursorEliminarUnion(ev)`

En caso de que la variable *dentro* sea **true** se procederá a borrar la unión seleccionada llamando al módulo *borrarUnionBD.php*. Este módulo va a eliminar las entradas de dicha unión en la base de datos y las imágenes de la unión del servidor. Este procedimiento de borrado va a ser prácticamente igual para eliminar las adaptaciones al usuario pero con dos diferencias.

```

$resultado = $enlace->query("DELETE FROM in_relationnodes WHERE idNodeInitial = '$idNodeInicial'
AND idNodeFinal = '$idNodeFinal' AND idMap = '$idMap' AND idUser = '$idUser';");

$resultado = $enlace->query("DELETE FROM in_relationnodes WHERE idNodeInitial = '$idNodeFinal'
AND idNodeFinal = '$idNodeInicial' AND idMap = '$idMap' AND idUser = '$idUser';");

unlink("Imagenes/Mapa_{$idMap}/$photo1");
unlink("Imagenes/Mapa_{$idMap}/$photo2");

```

Ilustración 35. Código de `borrarUnionBD.php`

La primera diferencia es que, al igual que para crear una unión adaptada hay que seleccionar un usuario de la lista desplegable de usuarios que hay en la sección *Adaptación al usuario*, para borrar una unión adaptada también hay que seleccionar el usuario del que se desea borrar la unión. Cuando se selecciona el usuario, se muestran las adaptaciones al usuario ya creadas, con lo que se podrá seleccionar una de ellas para borrarla.

La segunda diferencia es que no podemos usar la función *crearRectanguloUnion(Coords1, Coords2)*, ya que la línea de la unión adaptada es curva y no recta como la de la unión.

Por este motivo tendremos que usar la función *crearRectanguloLineaEspecial(Coords1, Coords2, {"x":arrayLineasD[i].puntoInt.x, "y":arrayLineasD[i].puntoInt.y})*, en la que aparte de pasarle por parámetros las coordenadas de los nodos de los extremos de la unión adaptada, también se pasan las coordenadas del punto de la curva de la línea.

6.2.3.3 Edición de uniones/adaptaciones al usuario

Para poder ver o editar una unión, lo primero que se deberá hacer es elegir la opción de edición clicando sobre el icono de un ojo que se encuentra en la sección de *Ver/Editar* de la barra de herramientas. Una vez que seleccionada esta opción, solo se deberá clicar sobre la unión o adaptación al usuario que se desea editar o ver al igual que cuando se quiere borrar una unión o unión adaptada. Para poder ver o editar las uniones adaptadas al usuario, primero se deberá seleccionar un usuario del desplegable que hay en la sección de *Adaptación al usuario*.

Con este clic sobre una unión se va a llamar al método *editarContenidos()*. Este método es el que decidirá si se ha pulsado sobre un unión o adaptación al usuario y cargará toda la información en un nueva pantalla mediante una llamada al método *mostrarModalUnión(pos)*. Este método mostrará una pantalla exactamente igual que la que se usa para guardar las imágenes cuando se quiere crear una nueva unión, pero con los

campos rellenos por los datos correspondientes. Para poder cargar la imagen del nodo, se realizará una llamada a ***obtenerFotosUnion.php***, que descargará del servidor las imágenes correspondientes a la unión seleccionada.

Una vez se tengan las imágenes cargadas en la pantalla de edición, se podrán modificar estas imágenes. Cuando se pulse sobre el botón *Guardar*, este llamará al método *comprobarCambiosUnion(pos, URL)*. Este método comprobará los cambios realizados y llamará a ***modificarUnionBD.php*** y a ***modificarImagenUniónServidor.php***. Estos módulos, a su vez modificarán las entradas en base de datos relativas a la unión y las imágenes de la unión.

7 Conclusiones y trabajo futuro

7.1 Conclusiones

El objetivo principal de este proyecto es el facilitar una herramienta web a los educadores de personas con discapacidad cognitiva que les permita crear caminos o rutas entre distintos puntos de un espacio interior.

La herramienta ha sido creada en una tecnología que más o menos todo el mundo ha usado alguna vez y con la que la mayoría de la población está familiarizada, como son las páginas web. La aplicación web permite la creación de caminos dividiendo estos en una secuencia de pasos a seguir hasta llegar al destino final del camino.

La interfaz de la aplicación ha sido creada entendiendo los problemas a los que se enfrentan los educadores, lo que ha permitido que esta sea se adecue al colectivo que la va a usar. La herramienta web ha sido implementada usando tecnologías que están la orden del día, lo que ha permitido obtener una aplicación web simple y sencilla, debido a que el rango de experiencia con las páginas web de las personas que la usen es muy amplio, a la par que usable.

Tras realizar un análisis exhaustivo de los problemas a los que se enfrentan los educadores, se han establecido una serie de requisitos que debe cumplir la aplicación. Para llegar a obtener una herramienta de calidad, se ha decidido usar un modelo en cascada con retroalimentación, el cual nos va a permitir establecer nuevos requisitos durante la implementación del proyecto sin poner en peligro la viabilidad de este.

La arquitectura que se ha utilizado para implementar esta aplicación ha sido una arquitectura cliente-servidor. El cliente únicamente se va a encargar de cargar la herramienta vía peticiones *http* sobre el servidor, donde se alojaran tanto los datos de la los mapas creados, como los módulos implementados y que dan componen la herramienta web.

7.2 Trabajo futuro

Las principales líneas de trabajo a seguir en un futuro para seguir mejorando la aplicación web son:

- Poder realizar pruebas directamente con educadores para que sean ellos quienes evalúen la aplicación y realicen un *feedback* sobre su experiencia con esta. De esto se sacarán una serie de conclusiones atendiendo a sus necesidades que permitirán mejorar la aplicación web y que se adapte a las capacidades de las personas que la usarán.

- Otra línea de trabajo no menos importante que la anterior será hacer compatible la aplicación compatible tanto para distintos *browsers* como para distintos dispositivos móviles (*smartphones*, *tablets*, etc). En el segundo caso se estudiará si es más coherente seguir manteniendo la herramienta web como página web, o si es preferible convertir esta en un app que facilite su uso en estos dispositivos.
- Una línea de trabajo muy importante es también marcar un plan de mantenimiento para la herramienta web, ya sea para comprobar el funcionamiento de la misma o para realizar un estudio de nuevas tecnologías que puedan permitir mejorar la herramienta mediante la implementación de proyectos evolutivos.

Referencias

- [1] F. Cegarra Andrés y G. García Villar. *Necesidades educativas especiales del alumnado con discapacidad intelectual* (Online).
<http://diversidad.murciaeduca.es/orientamur/gestion/documentos/unidad11.pdf>
- [2] American Association of Intellectual and Developmental Disabilities (AAIDD). *Definition of Intellectual Disability* (Online).
<http://aaidd.org/intellectual-disability/definition#.V0lkSvmLTIU>
- [3] Conferencia de Adolescencia y Juventud de Iberoamérica y El Caribe (CODIJAC). *DSM-5. Novedades y criterios diagnósticos* (Online).
<http://www.codajic.org/sites/www.codajic.org/files/DSM%205%20%20Novedades%20y%20Criterios%20Diagn%C3%B3sticos.pdf>
- [4] Ministerio de Industria, Energía y Turismo. *Proyectos de integración en las TIC de personas con discapacidad* (Online).
<http://www.red.es/redes/sala-de-prensa/reportaje/proyectos-de-integracion-en-las-tic-de-personas-con-discapacidad>
- [5] Wikipedia. *Metodología de desarrollo de software*. 2016. (Online).
https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software
- [6] Información sobre lenguaje HTML5.
<https://developer.mozilla.org/es/docs/HTML/HTML5>
- [7] Información sobre lenguaje Javascript.
https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript
- [8] Información sobre lenguaje JQuery.
<http://api.jquery.com/>
- [9] Información sobre lenguaje PHP.
<http://php.net/manual/es/index.php>
- [10] Información sobre formato JSON.
<http://json.org/>
- [11] Universidad Politécnica de Madrid (UPM). *Tecnologías y Servicios para la Sociedad de la Información*. (Online).
http://www.upm.es/sfs/Rectorado/Organos%20de%20Gobierno/Consejo%20Social/Actividades/tecnologias_servicios_para_sociedad_informacion.pdf
- [12] Servicio de sistemas de información geográfica y teledetección (SIGTE). Universidad de Girona. *Integración de un sistema de posicionamiento indoor en aplicaciones SIG para dispositivos móviles* (Online).
http://www.sigte.udg.edu/jornadassiglibre2013/uploads/articulos_13/a29.pdf
- [13] Miguel Michán. *Indoor Survey, Apple lanza una nueva app para posicionamiento en interiores*. (Online).
<http://www.applesfera.com/aplicaciones-ios-1/indoor-survey-apple-lanza-una-nueva-app-para-posicionamiento-en-interiores>